HySparK: Hybrid Sparse Masking for Large Scale Medical Image Pre-Training

Fenghe Tang[1,2], Ronghao Xu[1,2], Qingsong Yao[3], Xueming Fu[1,2], Quan Quan[3], Heqin Zhu[1,2], Zaiyi Liu[4,5], and S. Kevin Zhou[1,2,3] ✉

[1] School of Biomedical Engineering, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei, Anhui, 230026, P.R. China
[2] Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, Jiangsu, 215123, P.R. China
[3] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[4] Department of Radiology, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China
[5] Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China

Abstract. The generative self-supervised learning strategy exhibits remarkable learning representational capabilities. However, there is limited attention to end-to-end pre-training methods based on a hybrid architecture of CNN and Transformer, which can learn strong local and global representations simultaneously. To address this issue, we propose a generative pre-training strategy called **Hy**brid **Spar**se mas**K**ing (HySparK) based on masked image modeling and apply it to large-scale pre-training on medical images. First, we perform a bottom-up 3D hybrid masking strategy on the encoder to keep consistency masking. Then we utilize sparse convolution for the top CNNs and encode unmasked patches for the bottom vision Transformers. Second, we employ a simple hierarchical decoder with skip-connections to achieve dense multi-scale feature reconstruction. Third, we implement our pre-training method on a collection of multiple large-scale 3D medical imaging datasets. Extensive experiments indicate that our proposed pre-training strategy demonstrates robust transfer-ability in supervised downstream tasks and sheds light on HySparK's promising prospects. The code is available at https://github.com/FengheTan9/HySparK.

Keywords: Self-supervised learning · Masked image modeling · Hybrid architecture of CNN and Transformer · Medical images pre-training.

1 Introduction

Due to the scarcity of time-consuming and labor-intensive labeled medical images, pre-training on large amounts of easy-collected unlabeled medical images by self-supervised learning approaches to learn representations for down-

stream tasks is a promising approach in medical image analysis (MIA) [1]. Self-supervised learning approaches can be divided into two families: Contrastive methods [2,3,4,5,6,7] and Generative methods [8,9,10,11,12,13,14,15,16,17], where the latter group demonstrates better transferability to downstream tasks [13,17] such as segmentation. Representative generative methods like MAE [13] pre-train the Vision Transformers (ViTs) [25] in "BERT-style" [8] by dropping masked non-overlapping patches and re-predicting the masked ones.

From the architecture perspective, the inductive bias of CNN [24] and the long-range representation ability of Transformer [25] play pivotal roles in achieving excellent performance in visual tasks. However, the local limitation of CNNs constrains their ability to overcome performance bottlenecks further. Additionally, due to the scarcity and sparsity of medical images, the limited inductive biases and data-hungry nature of ViTs [25] make it challenging to effectively transfer to downstream tasks [18,30]. To integrate the advantages of both worlds at the infrastructure design level, a hybrid architecture leverages the inductive bias of CNNs and the global context learning capabilities of ViTs, showing great potential to break-through the performance bottlenecks on medical images [20,21,22,23].

Based on this advancement, a natural insight arises: Is it possible to simultaneously pre-train CNN and ViT with large-scale unlabeled medical images, which fully capitalize on the advantages of the hybrid model to unleash its potential? Despite MAE [13] is able to pre-train ViTs [25], for CNNs, executing sliding windows can erode the masked regions, leading to a vanishing mask pattern and causing a pixel distribution shift issue [17]. Luckily, SparK [17] successfully extends the masked image modeling to CNNs by deploying sparse convolution [39] to calculate only unmasked positions and skip the masked pixels.

Nevertheless, extending the success of the "BERT-style" masked imaging modeling pre-training strategy from single to hybrid architectures remains a challenging yet unrealized problem. Two main challenges are hindering the end-to-end implementation of pre-training hybrid architectures: **(i) Masking consistency.** For a single architecture, it is easy to maintain masking consistency [11,12] [13,14,16,17]. However, for hybrid architectures, due to the inconsistent masking strategies in both worlds, the direct combination still causes the "pixel distribution shift" and "mask pattern vanishing" issues [17]. **(ii) Multi-scale representation learning is necessary.** In medical imaging, a series of u-shape networks such as U-Net [24] demonstrate the importance of multi-scale and skip-connections in improving model performance. However, most current algorithms only learn representations at a single scale [13,14], neglecting the performance advantages brought by multi-scale architectures, which is crucial in MIA tasks. Although SparK [17] takes this issue into account, it uses a simple fusion method (only skip-addition) and ignores important pattern adaptation in downstream tasks (success of skip-connections in medical downstream tasks [24,26,29,30]), which widen the gap from pre-training to downstream transferring.

In this work, we address the above issues and propose a **Hy**brid **Spar**se mas**K**ing (HySparK) strategy for self-supervised learning in CNN and Trans-

former hybrid architectures. Following the success of hybrid architectures in MIA tasks [20,21,22,23], we use CNN as the top encoder to extract local representations and Transformer as the bottom for global features. Moreover, skip-connections are introduced to integrate multi-scale representations. To address **masking consistency**, we perform **bottom-up** masking. Specifically, we initialize the masks based on the patch division in the bottom ViT layer, which are then mapped to the upper CNN layers with different scales. Then, we use sparse convolution in CNN and drop masked patches in Transformer to avoid calculating mask regions. This novel design ensures the consistency of mask mapping between different layers in the hierarchical CNN and ViT, which prevents the data distribution shift problem [17] in the hybrid encoder.

In the decoding stage, to leverage the advantage of skip-connections mentioned in issue **(ii)**, we construct a simple hierarchical decoder. For the skip-connections (concat and fuse), we fill the mask embeddings into all empty positions of multi-scale features. Finally, we reconstruct the masked pixels. To our knowledge, HySparK is the first successful generative-based 3D hybrid architecture method for self-supervised learning, applied to large-scale 3D CT medical image pre-training. Similar to SparK [17], HySparK is a general method that does not restrict the specific hybrid encoder (e.g. specific CNN or Transformer) to pre-train. In this paper, we utilize the representative CNN network in the medical imaging analysis, modern MedNeXt [26], as the top CNN. For the bottom Transformer, we employ the standard ViT [25]. Across multiple segmentation downstream tasks, HySparK outperforms state-of-the-art medical self-supervised pre-training methods and single-architecture-based masking approaches like MAE [13]. Our primary contributions are as follows:

1. We propose a generative self-supervised learning method to **pre-train a hybrid architecture**, which unleashes its strengths to integrate both local and global representations. We are the **first** to pre-train the hybrid architecture in an end-to-end fashion.
2. We design the bottom-up **3D hybrid masking** to keep the consistency of mask modeling and data distribution across different architectures.
3. We pre-train a strong hybrid vision encoder using HySparK on large-scale CT medical image datasets (6.8K CT scans in total). Extensive experiments on downstream tasks demonstrate the **effectiveness and potential** of HySparK in its transferability.

2 Approach

As shown in Fig. 1, our proposed HySparK framework aims to mask a portion of the image through hybrid masking and pre-train the encoder by reconstructing the masked patches. The HySparK framework comprises two stages: the CNN stage and the Transformer stage. Firstly, the bottom-up sparse masking is performed in the encoder (Section 2.1), where hierarchical 3D encoding is conducted in the upper CNN stage, while patch-based 3D encoding takes place in

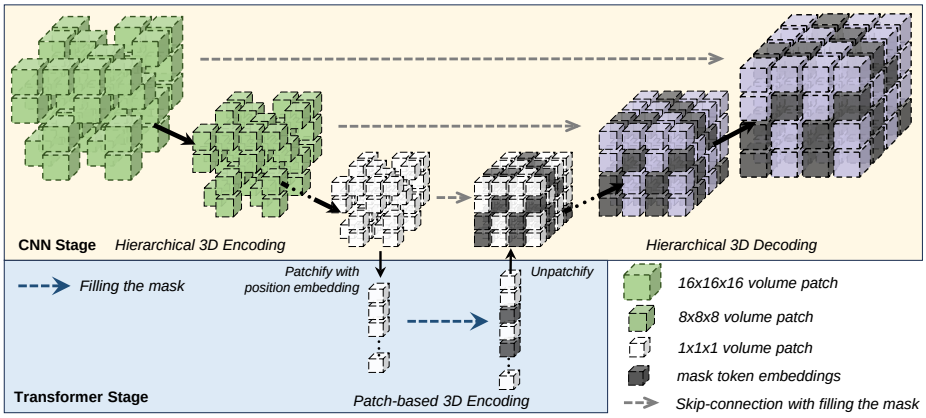


Fig. 1. Hybrid Sparse masKing (HySparK). The hybrid architecture comprises a CNN at the top (yellow) and a Transformer at the bottom (blue). We initiate the masking strategy at the junction between the CNN and Transformer and execute bottom-up mask modeling. The initialization unmasking patch is white, the bottom-up mapping unmasking patch is green and the masking position is **black**.

the bottom Transformer stage. Secondly, hierarchical 3D decoding is conducted using a hierarchical decoder with skip-connections (Section 2.2) to learn multi-scale representations. Finally, we describe the pre-train optimization objectives of HySparK (Section 2.3).

2.1 Hybrid masking

We perform hybrid masking in a bottom-up manner. Specifically, we divide the encoder into a top-level N-stage CNN encoder $[E^{cnn}_{1\sim N}]$ (*e.g.*, $N = 4$ stages ResNet-style [27] or ConvNeXt-style [28] encoder) and a bottom-level ViT encoder E^{tr}. We initialize sparse mask $M_n \in \mathbb{R}^{\frac{H}{n^2}\times\frac{W}{n^2}\times\frac{D}{n^2}}$ at the junction of the two architectures (i.e. initializing masking at the output of the last CNN layer, before ViT). To maintain consistency in masking across different architectures, we ensure that both architectures adhere to the junction-initialized masking.

Hierarchical 3D encoding in CNN stage. Since the masking is initialized at the last layer of the CNN, we upsample the initialized M_n sparsely backward to different CNN stages, generating a set of mask $[M_{1\sim n-1}]$ at different scales with the same rules from M_n. Subsequently, we utilize 3D sparse convolutions to generate different scales sparse features $[S_{1\sim n}]$ with masking $[M_{1\sim n}]$ and feature maps $[f^{cnn}_{1\sim n}]$:

$$\text{SparseConv}(f^{cnn}_i, M_i) \rightarrow S_i, \quad \forall i \in \{1, 2, ..., N\}. \tag{1}$$

Patch-based 3D encoding in Transformer stage. As the bottom encoder is a standard ViT, learning only unmasked patches. We divide the features obtained from the last layer output of the CNN into patches with position embeddings.

Subsequently, following the initialized masking rules, we remove the masked patches and only utilize the tokens T without masking:

$$\text{Patchify}(S_n, M_n) \rightarrow T. \tag{2}$$

2.2 Hierarchical decoding with skip connections

In decoding stage, we introduce a simple cascaded decoder comprising $N-1$ upsampling blocks $\{B_1^{up}, B_2^{up}, ..., B_{n-1}^{up}\}$ and $N-1$ fusion blocks $\{B_1^{f}, B_2^{f}, ..., B_{n-1}^{f}\}$ for skip connections. Before decoding, we first unpatchify the tokens T from the Transformer stage into sparse feature map S_n. Next, we fill mask embeddings into all empty positions of the sparse features at different scales to get dense features $[S_{1\sim n}^{'}]$. After applying projection layers to reduce the width of dense features at different scales, we perform hierarchical decoding with skip connection via:

$$D_n = \phi_n(S_n^{'}). \tag{3}$$

$$D_i = B_i^{f}(\text{Concat}\{B_i^{up}(D_{i+1}), \phi_i(S_i^{'})\}), \quad (\forall i \in \{N-1, ..., 2, 1\}). \tag{4}$$

where ϕ_i denotes the linear projection layer of the ith stage. D_n and D_i represent the input and output of the decoder. The final output of the decoder is D_1.

2.3 Optimization objectives and downstream fine-tuning

We utilize a linear layer to reconstruct D_1. Moreover, similar to MAE [13] and SparK [17], a mean square error loss ($\mathcal{L}_2$) is used for reconstruction optimization of normalized pixels at masked positions. During fine-tuning, we only use the encoder to accomplish downstream tasks without any adjustment, as dense input is a special case of sparse input [17]. We use a combined loss ($\mathcal{L}_{seg}$) of binary cross entropy ($\mathcal{L}_{BCE}$) and Dice loss ($\mathcal{L}_{Dice}$) to optimize the network.

3 Experiment

3.1 Datasets

Pre-training datasets: A total of 13 public CT datasets, consisting of **6,814** CT scans, are curated to form our pre-training dataset (reviewed in Table 1). Existing annotations or labels are not utilized from these datasets during pre-training. The pre-train datasets are interpolated to the isotropic voxel spacing of 1.5 mm. Intensities are scaled to $[-175, 250]$, then normalized to $[0, 1]$. We crop sub-volumes of $96 \times 96 \times 96$ voxels.

BTCV dataset: The BTCV dataset [31] consists of 30 subjects with abdominal CT scans where 13 organs are annotated by interpreters under supervision of clinical radiologists at Vanderbilt University Medical Center. Our data preprocessing strategy is the same as UNETR [29].

MSD datasets: Medical Segmentation Decathlon (MSD) dataset [38] comprises ten segmentation tasks from different organs and image modalities. We only use six CT datasets: Liver, Lung, Pancreas, Hepatic Vessel, Spleen, and Colon datasets. All the pre-processing strategies are the same as Swin UNETR [30].

Table 1. Overview of Pre-train Dataset.

Dataset (year)	# of classes	# of volumes	downstream	Dataset (year)	# of classes	# of volumes	downstream
BTCV (2015) [31]	13	50	✓	MSD Liver (2021) [38]	2	201	✓
CHAOS (2018) [32]	4	40		MSD Lung (2021) [38]	2	95	✓
WORD (2021) [33]	16	150		MSD Pancreas (2021) [38]	2	420	✓
FLARE'22 (2022) [34]	13	2300		MSD Hepatic Vessel (2021) [38]	1	443	✓
AbdomenCT-1k (2022) [35]	4	1062		MSD Spleen (2021) [38]	1	61	✓
TotalSegmentator (2022) [36]	104	1202		MSD Colon (2021) [38]	1	190	✓
AMOS22 (2022) [37]	15	600		Total		6814	

3.2 Settings

HySparK can use any 3D convolutional network and patch-based ViT as the hybrid architecture's encoder. In the CNN stage, we choose MedNeXt [26] (the state-of-the-art ConvNet in medical tasks) as the top encoder. In the Transformer stage, we implement the standard ViT [25] as the bottom encoder. It is worth noting that we substitute the downsampling layers of MedNeXt with max-pooling. Additionally, for the pre-trained decoder, the upsampling block consists of two convolutional layers and an upsampling layer, while the fusion block comprises two convolutional layers. For downstream tasks, we utilize the MedNeXt decoder for segmentation.

For pre-training tasks, we train with an AdamW optimizer, an initial learning rate of 1e-4, and a cosine-annealing learning rate scheduler. The pre-training experiments use a batch-size of 8 on a single GPU and 100 epochs in 4 days. For downstream segmentation tasks, a five-fold cross-validation strategy is used to train models for all BTCV and MSD experiments and we select the best model in each fold. Detailed training hyperparameters for fine-tuning BTCV and MSD tasks are the same as Swin UNETR [30]. All methods are implemented in PyTorch and trained on an Nvidia A800.

The Dice similarity coefficient (Dice) is used as the measurement for experiment results. We select three advanced generative-based self-supervised learning strategies: Transformer-based MAE [13] and SimMIM [13], CNN-based SparK [17] and two advanced contrastive-based self-supervised learning method: Swin UNETR [30] Pre-trained method (SUP) and vox2vec [19]. In addition, we choose the current well-known segmentation networks UNETR [29], Swin UNETR [30], and MedNeXt [26] as the downstream segmentation task networks of MAE, SimMIM, and SparK, respectively. It is worth noting that MAE, SimMIM, and SparK methods are obtained by using official codes and extending them to 3D.

3.3 Results and discussion

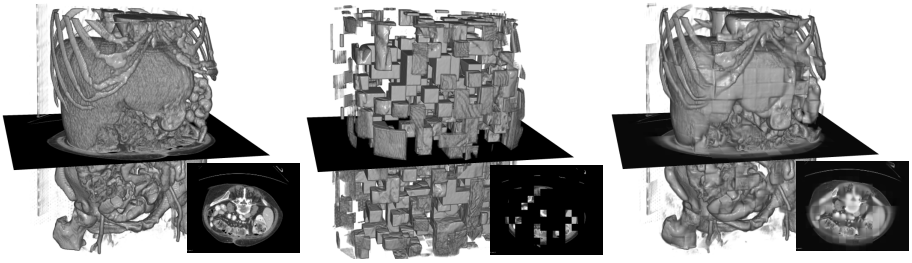
Results on BTCV dataset. Evaluation results on BTCV are shown in Table 2. Compared with other competitive methods, the proposed HySparK achieves the best performance. We obtain the highest average Dice of 80.67%, which at least improves by 1.17% compared to other baselines. Additionally, we achieve significant improvements in segmenting organs with smaller sizes, such as the pancreas and adrenal glands. This shows that HySparK effectively learns strong multi-scale representations. In addition, we fine-tune the pre-trained models using a

Table 2. Result on BTCV. **val** (bold) / val (underline) : top method / second method.

	Pre-training Method		Spl	Kid	Gall	Eso	Liv	Sto	Aor	IVC	Veins	Pan	AG	Avg
	Method	Network												
20%	vox2vec [19]	3D UNet(FPN) [24]	73.97	66.80	34.64	49.51	86.72	54.28	73.70	64.43	41.64	37.35	27.00	54.14
	SUP [30]	Swin UNETR [30]	71.29	59.78	40.43	57.30	87.91	51.24	70.88	59.13	50.16	37.70	34.24	54.93
	MAE [13]	UNETR [29]	66.22	**78.63**	**48.19**	24.02	**92.51**	**78.50**	79.00	**78.28**	35.63	52.99	18.52	57.91
	SimMIM [13]	Swin UNETR [30]	71.16	62.76	42.32	56.05	88.45	52.46	71.95	60.94	51.32	39.39	35.10	56.14
	SparK [17]	MedNeXt [26]	79.78	72.29	38.66	58.89	91.71	68.68	**81.24**	71.93	57.13	51.96	29.01	61.74
	HySparK	MedNeXt+ViT	**79.84**	70.54	46.03	**60.86**	91.29	67.12	79.42	69.21	**59.07**	**54.34**	**43.61**	**64.27**
100%	vox2vec [19]	3D UNet(FPN) [24]	**91.40**	**90.70**	59.50	72.70	**96.30**	83.20	**91.30**	83.90	69.20	73.90	65.20	79.50
	SUP [30]	Swin UNETR [30]	84.20	86.70	58.40	70.40	94.50	76.00	87.70	82.10	67.00	69.80	61.00	75.80
	MAE [13]	UNETR [29]	90.71	87.63	62.50	70.69	94.73	86.11	90.59	83.26	**71.00**	75.47	63.77	79.07
	SimMIM [13]	Swin UNETR [30]	87.12	80.85	60.28	72.34	93.70	78.42	87.89	81.46	64.92	66.34	58.65	74.73
	SparK [17]	MedNeXt [26]	90.02	87.78	62.48	**74.36**	95.00	84.85	90.17	83.60	68.83	76.57	64.13	79.21
	HySparK	MedNeXt+ViT	90.67	88.32	**68.18**	74.20	95.03	**87.46**	90.17	**84.50**	70.04	**78.36**	**66.75**	**80.67**

Table 3. Result on MSD. **val** (bold) / val (underline) : top method / second method.

Pre-training method		Liver			Lung	Pancreas			Hepatic Vessel			Spleen	Colon	Avg
Method	Network	Dice1	Dice2	Avg	Dice	Dice1	Dice2	Avg	Dice1	Dice2	Avg	Dice	Dice	
vox2vec [19]	3D UNet(FPN) [24]	95.60	51.00	73.70	56.60	77.00	31.80	54.40	59.50	62.40	60.95	96.10	30.10	61.97
SUP [29]	Swin UNETR [29]	95.00	49.30	72.15	55.20	75.20	35.90	55.55	60.90	57.50	59.20	95.50	29.20	61.13
MAE [13]	UNETR [29]	95.49	56.47	75.98	56.42	77.76	39.29	58.52	59.99	62.22	61.10	95.28	34.53	63.63
SimMIM [13]	Swin UNETR [29]	95.32	55.25	75.28	60.31	76.16	44.96	60.56	60.67	61.79	61.23	95.64	41.11	65.68
SparK [17]	MedNeXt [26]	95.87	**62.95**	**79.41**	65.58	78.88	47.86	63.37	61.08	67.76	64.42	96.18	49.85	69.80
HySparK	MedNeXt+ViT	**96.02**	60.92	78.47	**65.96**	**79.69**	**49.67**	**64.68**	**61.58**	**69.36**	**65.47**	**96.39**	**50.78**	**70.29**



(a) Raw Input **(b)** Masked Input **(b)** Prediction

Fig. 2. Reconstruction Result by HySparK.

smaller (20%) training set, our HySparK significantly outperforms state-of-the-art methods (average Dice of 64.27, 2.5% higher than other methods) and gains the best trade-off performance in different scale organs, which highlights the powerful downstream transferring capability of our method.

Results on MSD datasets. The overall results on the MSD dataset per task are shown in Table 3. HySparK presents the best average Dice of 70.29%. Our method outperforms other SOTA approaches in Lung, Pancreas, Hepatic Vessel, Spleen, and Colon tasks. Moreover, HySparK improves Pancreas and Hepatic Vessel lesions by at least 1.81% and 1.60% which is attributed to its strong multi-scale representation. It is worth noting that almost all generative-based methods outperform contrast-based methods, indicating the superior transferability of generative-based methods.

Table 4. Ablation study on Mask Ratio.

Mask Ratio	Spl	RKid	Lkid	Gall	Eso	Liv	Sto	Aor	IVC	Veins	Pan	Rad	Lad	BTCV Avg
w/o pre-trained	89.71	88.17	86.69	62.73	73.14	94.44	83.96	88.94	82.51	70.02	72.47	64.73	63.99	78.58
mask 25 %	90.52	89.76	87.55	66.42	74.93	94.98	86.21	90.61	84.28	70.96	77.92	66.26	66.90	80.56
mask 50 %	90.76	88.46	86.33	65.38	75.50	95.21	85.43	90.75	83.57	71.75	77.47	66.09	68.02	80.36
mask 75 %	90.67	89.35	87.30	68.18	74.20	95.03	87.46	90.17	84.50	70.04	78.36	66.46	67.04	80.67

Table 5. Ablation study on each components in HySparK.

HySparK components	hybrid masking	skip-connection	skip-addition	BTCV Avg
w/o pre-trained	—	—	—	78.58
w/o bottom-up masking	✗	✓	✗	79.47
w/o skip	✓	✗	✗	78.80
w/ skip-addition	✓	✗	✓	79.97
HySparK	✓	✓	✗	80.67

Visualization. We visualize 3D reconstruction results to check what HySparK learns in pre-training. As shown in Fig. 2, our method can almost reconstruct the different shapes of organs, bones, and other details from the very small portion of unmasked patches.

3.4 Ablation study

Ablation Study on Mask Ratio. Table 4 shows the influence of different mask ratios on the model. Surprisingly, similar to MAE [13], it can be found that a 75% mask ratio achieves the best performance in downstream tasks.
Ablation study on HySparK components. We first remove the two most important designs in HySparK: bottom-up hybrid masking and skip connections. When mask consistency is not maintained, we observe a significant performance degradation in row 2 of Table 5 that almost reaches the vanilla model (row 1). It suggests that *inconsistency masking will lead to ineffective pre-training*. We then remove the skip design (row 3) or only use skip-addition (row 4), the performance drops significantly compared to using skip-connections (row 5), which illustrates the importance of pattern alignment for pre-training and fine-tuning tasks.
Ablation study on architecture. As demonstrated in Table 2 and 3, when the hybrid architecture drops to single architecture (i.e., CNN in SparK or ViT in MAE), their performance experiences a certain decrease compared to the hybrid architecture. This demonstrates the significant role of the hybrid architecture and its masking strategy in medical image tasks.

4 Conclusion

The success of hybrid architectures in medical tasks prompts us to explore their potential in downstream tasks after being well pre-trained using large-scale unlabeled medical images. In this paper, we introduce HySparK, a generative self-supervised approach to pre-training hybrid architectures, which creates a

bottom-up masking modeling strategy to solve the masking inconsistency. For the problem of data distribution shift, we use sparse convolution for encoding in the CNN stage and predict the masked tokens using unmasked patches in the Transformer stage. Moreover, we introduce skip-connections to achieve pre-training and downstream task pattern alignment. HySparK brings significant performance leaps in downstream tasks and we hope our findings can inspire more work to maximize the potential of hybrid architectures in medical tasks.

Acknowledgments. Supported by Natural Science Foundation of China under Grant 62271465, Suzhou Basic Research Program under Grant SYG202338, and Open Fund Project of Guangdong Academy of Medical Sciences, China (No. YKY-KF202206).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Zhang C, Zheng H, Gu Y. Dive into the details of self-supervised learning for medical image analysis[J]. Medical Image Analysis, 2023, 89: 102879.
2. Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//ICML. PMLR, 2020: 1597-1607.
3. He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//CVPR. 2020: 9729-9738.
4. Grill J B, Strub F, Altché F, et al. Bootstrap your own latent-a new approach to self-supervised learning[J]. NeurIPS, 2020, 33: 21271-21284.
5. Caron M, Misra I, Mairal J, et al. Unsupervised learning of visual features by contrasting cluster assignments[J]. NeurIPS, 2020, 33: 9912-9924.
6. Chen X, He K. Exploring simple siamese representation learning[C]//CVPR. 2021: 15750-15758.
7. Caron M, Touvron H, Misra I, et al. Emerging properties in self-supervised vision transformers[C]//ICCV. 2021: 9650-9660.
8. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
9. Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. NeurIPS, 2020, 33: 1877-1901.
10. Pathak D, Krahenbuhl P, Donahue J, et al. Context encoders: Feature learning by inpainting[C]//CVPR. 2016: 2536-2544.
11. Bao H, Dong L, Piao S, et al. Beit: Bert pre-training of image transformers[J]. arXiv preprint arXiv:2106.08254, 2021.
12. Zhou J, Wei C, Wang H, et al. ibot: Image bert pre-training with online tokenizer[J]. arXiv preprint arXiv:2111.07832, 2021.
13. He K, Chen X, Xie S, et al. Masked autoencoders are scalable vision learners[C]//CVPR. 2022: 16000-16009.
14. Xie Z, Zhang Z, Cao Y, et al. Simmim: A simple framework for masked image modeling[C]//CVPR. 2022: 9653-9663.
15. Assran M, Duval Q, Misra I, et al. Self-supervised learning from images with a joint-embedding predictive architecture[C]//CVPR. 2023: 15619-15629.
16. Chen X, Ding M, Wang X, et al. Context autoencoder for self-supervised representation learning[J]. IJCV, 2024, 132(1): 208-223.

17. Tian K, Jiang Y, Diao Q, et al. Designing bert for convolutional networks: Sparse and hierarchical masked modeling[J]. arXiv preprint arXiv:2301.03580, 2023.
18. Zhou L, Liu H, Bae J, et al. Self pre-training with masked autoencoders for medical image classification and segmentation[C]//ISBI. IEEE, 2023: 1-6.
19. Goncharov M, Soboleva V, Kurmukov A, et al. vox2vec: A framework for self-supervised contrastive learning of voxel-level representations in medical images[C]//MICCAI, 2023: 605-614.
20. Chen J, Lu Y, Yu Q, et al. Transunet: Transformers make strong encoders for medical image segmentation[J]. arXiv preprint arXiv:2102.04306, 2021.
21. Wang W, Chen C, Ding M, et al. Transbts: Multimodal brain tumor segmentation using transformer[C]//MICCAI, 2021: 109-119.
22. Wang, Haonan, et al. Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. AAAI. Vol. 36. No. 3. 2022.
23. Tang F, Nian B, Ding J, et al. MobileUtr: Revisiting the relationship between lightweight CNN and Transformer for efficient medical image segmentation[J]. arXiv preprint arXiv:2312.01740, 2023.
24. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//MICCAI, 2015: 234-241.
25. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.
26. Roy S, Koehler G, Ulrich C, et al. Mednext: transformer-driven scaling of convnets for medical image segmentation[C]//MICCAI, 2023: 405-415.
27. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//CVPR. 2016: 770-778.
28. Liu Z, Mao H, Wu C Y, et al. A convnet for the 2020s[C]//CVPR. 2022: 11976-11986.
29. Hatamizadeh A, Tang Y, Nath V, et al. Unetr: Transformers for 3d medical image segmentation[C]//WACV. 2022: 574-584.
30. Tang Y, Yang D, Li W, et al. Self-supervised pre-training of swin transformers for 3d medical image analysis[C]//CVPR. 2022: 20730-20740.
31. Landman B, Xu Z, Igelsias J, et al. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge[C]//Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge. 2015, 5: 12.
32. Kavur A E, Gezer N S, Barış M, et al. CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation[J]. Medical Image Analysis, 2021, 69: 101950.
33. Luo X, Liao W, Xiao J, et al. WORD: A large scale dataset, benchmark and clinical applicable study for abdominal organ segmentation from CT image[J]. Medical Image Analysis, 2022, 82: 102642-102642.
34. Ma J, Zhang Y, Gu S, et al. Unleashing the strengths of unlabeled data in pan-cancer abdominal organ quantification: the flare22 challenge[J]. arXiv preprint arXiv:2308.05862, 2023.
35. Ma J, Zhang Y, Gu S, et al. Abdomenct-1k: Is abdominal organ segmentation a solved problem?[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(10): 6695-6714.
36. Wasserthal J, Breit H C, Meyer M T, et al. Totalsegmentator: Robust segmentation of 104 anatomic structures in ct images[J]. Radiology: Artificial Intelligence, 2023, 5(5).
37. Ji Y, Bai H, Ge C, et al. Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation[J]. NeurIPS, 2022, 35: 36722-36732.

38. Antonelli M, Reinke A, Bakas S, et al. The medical segmentation decathlon[J]. Nature communications, 2022, 13(1): 4128.
39. Graham B, Van der Maaten L. Submanifold sparse convolutional networks[J]. arXiv preprint arXiv:1706.01307, 2017.