Medical Image Analysis 95 (2024) 103159

Contents lists available at ScienceDirect

Medical Image Analysis

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

Stepwise incremental pretraining for integrating discriminative, restorative, and adversarial learning

Zuwei Guo [a], Nahid Ul Islam [a], Michael B. Gotway [b], Jianming Liang [a,*]

[a] Arizona State University, Tempe, AZ 85281, USA
[b] Mayo Clinic, Scottsdale, AZ 85259, USA

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Self-supervised learning
Discriminative learning
Restorative learning
Adversarial learning
United framework
Stepwise pretraining

ABSTRACT

We have developed a *United* framework that integrates three self-supervised learning (SSL) ingredients (discriminative, restorative, and adversarial learning), enabling collaborative learning among the three learning ingredients and yielding three transferable components: a discriminative encoder, a restorative decoder, and an adversary encoder. To leverage this collaboration, we redesigned nine prominent self-supervised methods, including Rotation, Jigsaw, Rubik's Cube, Deep Clustering, TransVW, MoCo, BYOL, PCRL, and Swin UNETR, and augmented each with its missing components in a United framework for 3D medical imaging. However, such a United framework increases model complexity, making 3D pretraining difficult. To overcome this difficulty, we propose stepwise incremental pretraining, a strategy that *unifies* the pretraining, in which a discriminative encoder is first trained via discriminative learning, the pretrained discriminative encoder is then attached to a restorative decoder, forming a skip-connected encoder–decoder, for further joint discriminative and restorative learning. Last, the pretrained encoder–decoder is associated with an adversarial encoder for final full discriminative, restorative, and adversarial learning. Our extensive experiments demonstrate that the stepwise incremental pretraining stabilizes United models pretraining, resulting in significant performance gains and annotation cost reduction via transfer learning in six target tasks, ranging from classification to segmentation, across diseases, organs, datasets, and modalities. This performance improvement is attributed to the synergy of the three SSL ingredients in our United framework unleashed through stepwise incremental pretraining. Our codes and pretrained models are available at GitHub.com/JLiangLab/StepwisePretraining.

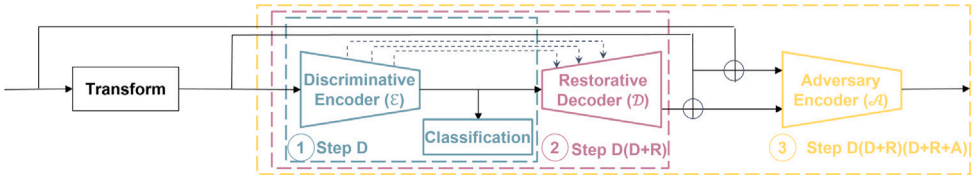
1. Introduction

Self-supervised learning (SSL) (Jing and Tian, 2020) pretrains generic source models (Zhou et al., 2021b) without using expert annotation, allowing the pretrained generic source models to be quickly fine-tuned into high-performance application-specific target models to minimize annotation cost (Tajbakhsh et al., 2021). The existing SSL methods typically employ one or a combination of the following three learning ingredients (Haghighi et al., 2022): (1) discriminative learning, which pretrains an encoder by distinguishing images associated with (computer-generated) pseudo labels; (2) restorative learning, which pretrains an encoder–decoder by reconstructing original images from their distorted versions; and (3) adversarial learning, which pretrains an additional adversary encoder to enhance restorative learning. It has already been demonstrated in Haghighi et al. (2021) that combining self-supervised discriminative methods and restoration enhances network performance in both classification and segmentation tasks. Further, (Tao et al., 2020) demonstrated that reconstructive method is further enhanced by adversarial learning. Inspired by both (Haghighi et al., 2021; Tao et al., 2020), we believe that combining all three components – discriminative, restorative, and adversarial learning – yields the best performance. Haghighi et al. (2022, 2024) articulated a vision and insights for integrating three learning ingredients in one single framework for collaborative learning, yielding three learned components: a discriminative encoder, a restorative decoder, and an adversary encoder (Fig. 1). However, such integration inevitably increases model complexity and pretraining difficulty, raising these two questions: *(a) how to optimally pretrain such complex generic models, and (b) how to effectively utilize pretrained components for target tasks?*

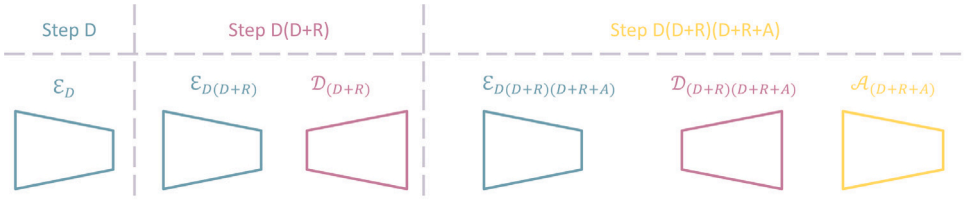
To address these two questions, we have redesigned nine prominent SSL methods for 3D imaging, including Rotation (Gidaris et al., 2018), Jigsaw (Noroozi and Favaro, 2016), Rubik's Cube (Zhuang et al., 2019), Deep Clustering (Caron et al., 2018), TransVW (Haghighi et al., 2021), MoCo (Momentum Contrast) (He et al., 2020), BYOL (Bootstrap Your Own Latent) (Grill et al., 2020), PCRL (Preservational

* Corresponding author.
E-mail address: Jianming.Liang@asu.edu (J. Liang).

https://doi.org/10.1016/j.media.2024.103159
Received 22 May 2023; Received in revised form 17 December 2023; Accepted 22 March 2024
Available online 16 April 2024



(a) Our *United* model with stepwise incremental pretraining: D(D+R)(D+R+A)



(b) Pretrained components resulted from stepwise incremental pretraining strategy: D(D+R)(D+R+A)

Fig. 1. Our *United* model consists of three components: a discriminative encoder $\mathcal{E}$, a restorative decoder $\mathcal{D}$, and an adversary encoder $\mathcal{A}$, where the discriminative encoder and the restorative decoder are skip connected, forming an encoder–decoder $(\mathcal{E}, D)$. To overcome the United model complexity and pretraining difficulty, we develop a strategy, called D(D+R)(D+R+A), to incrementally train the three components in a stepwise fashion: (1) Step D trains a discriminative encoder $\mathcal{E}_\emptyset$, where $\emptyset$ indicates that Encoder $\mathcal{E}$ is randomly initialized, via discriminative learning (*i.e.*, D), leading to a pretrained discriminative encoder $\mathcal{E}_D$; (2) Step D(D+R) attaches the pretrained discriminative encoder $\mathcal{E}_D$ to a randomly-initialized restorative decoder $D_\emptyset$ for further joint discriminative and restorative learning (*i.e.*, D+R), yielding a pretrained discriminative encoder $\mathcal{E}_{D(D+R)}$ and a pretrained restorative decoder $D_{(D+R)}$; (3) Step D(D+R)(D+R+A) associates the pretrained encoder–decoder $(\mathcal{E}_{D(D+R)}, D_{(D+R)})$ with a randomly-initialized adversarial encoder $\mathcal{A}_\emptyset$ for final full discriminative, restorative, and adversarial learning (*i.e.*, D+R+A), resulting in a pretrained discriminative encoder $\mathcal{E}_{D(D+R)(D+R+A)}$, a pretrained restorative decoder $\mathcal{E}_{(D+R)(D+R+A)}$, and a pretrained adversarial encoder $\mathcal{A}_{(D+R+A)}$. This stepwise incremental pretraining has proven to be reliable across multiple SSL methods (Fig. 2) for a variety of target tasks across diseases, organs, datasets, and modalities.

Contrastive Representation Learning) (Zhou et al., 2021a), and Swin UNETR (Swin UNEt TRansformers) (Tang et al., 2022). Among these methods, Rotation, Jigsaw, and Rubik's Cube are classic discriminative methods. Deep Clustering is a classic clustering method. TransVW and PCRL are methods that integrate both discriminative and restorative approaches. MoCo and BYOL are contrastive methods. Swin UNETR is a transformer-based model that incorporates contrastive, restorative, and discriminative methods. With these methods, we aim to encompass all components and models of SSL, emphasizing the generality of our approach. We formulated each methods in a single framework called "United" (Fig. 2), as it unites discriminative, restorative, and adversarial learning. Pretraining United models, with all three components together, directly from scratch is unstable (Table 2); therefore, we have investigated various training strategies and discovered a stable solution: stepwise incremental pretraining. Such pretraining is accomplished as follows: first training a discriminative encoder via discriminative learning, called Step D, then attaching the pretrained discriminative encoder to a restorative decoder (i.e., forming an encoder–decoder) for further combined discriminative and restorative learning, called Step D(D+R), and finally associating the pretrained autoencoder with an adversarial-encoder for the final full discriminative, restorative, and adversarial training, called Step D(D+R)(D+R+A). This stepwise pretraining strategy provides the most reliable performance across most target tasks evaluated in this work encompassing both classification and segmentation (see Table 2, 3, 4, 5, and 7).

Through our extensive experiments, we have observed that (1) discriminative learning alone (i.e., Step D) significantly enhances discriminative encoders on target classification tasks (e.g., +4%, 6%, and 1% AUC (Area Under the ROC Curve) improvement for lung nodule, pulmonary embolism and pulmonary embolism with vessel-oriented image representation false positive reduction as shown in Table 3) relative to training from scratch; (2) in comparison with (sole) discriminative learning, incremental restorative pretraining combined with continual discriminative learning (i.e., Step D(D+R)) enhances discriminative encoders further for target classification tasks (e.g., +2%, +4%, and +2% AUC improvement for lung nodule, pulmonary embolism and pulmonary embolism with vessel-oriented image representation false positive reduction as shown in Table 3) and boosts encoder–decoder models for target segmentation tasks (e.g., +3%, +7%, and +5% IoU (Intersection over Union) improvement for lung nodule, liver, and brain tumor segmentation as shown in Table 5); and (3) compared with Step D(D+R), the final stepwise incremental pretraining (i.e., Step D(D+R)(D+R+A)) generates sharper and more realistic medical images (e.g., FID decreases from 427.6 to 251.3 as shown in Table 6) and further strengthens each component for representation learning, leading to considerable performance gains (see Fig. 4) and annotation cost reduction (e.g., 28%, 43%, and 26% faster for lung nodule false positive reduction, lung nodule tumor segmentation, and pulmonary embolism false positive reduction as shown in Fig. 5) for six target tasks across diseases, organs, datasets, and modalities.

We should note that recently (Haghighi et al., 2022) also combined discriminative, restorative, and adversarial learning, but our findings complement theirs, and more importantly, our method significantly differs from theirs, because they were more concerned with contrastive learning (e.g., MoCo-v2 (Chen et al., 2020), Barlow Twins (Zbontar et al., 2021), and SimSiam (Chen and He, 2021)) and focused on 2D medical image analysis. By contrast, we are focusing on 3D medical imaging by redesigning nine popular SSL methods beyond contrastive learning. As they acknowledged (Haghighi et al., 2022), their results on TransVW (Haghighi et al., 2021) augmented with an adversarial encoder were based on the experiments presented in this paper. Furthermore, this paper focuses on a stepwise incremental pretraining to stabilize United model training, revealing new insights into synergistic effects and contributions among the three learning ingredients.

In summary, we make the following three main contributions:

1. A stepwise incremental pretraining strategy that stabilizes United models' pretraining and releases the synergistic effects of the three SSL ingredients;
2. A collection of pretrained United models that integrate discriminative, restorative, and adversarial learning in a single framework for 3D medical imaging, encompassing both classification and segmentation tasks, and;
3. A set of extensive experiments that demonstrate how various pretraining strategies benefit each SSL method for target tasks across diseases, organs, datasets, and modalities.

2. United framework and stepwise incremental pretraining

We have redesigned nine prominent SSL methods, including Rotation, Jigsaw, Rubik's Cube, Deep Clustering, TransVW, MoCo, BYOL,

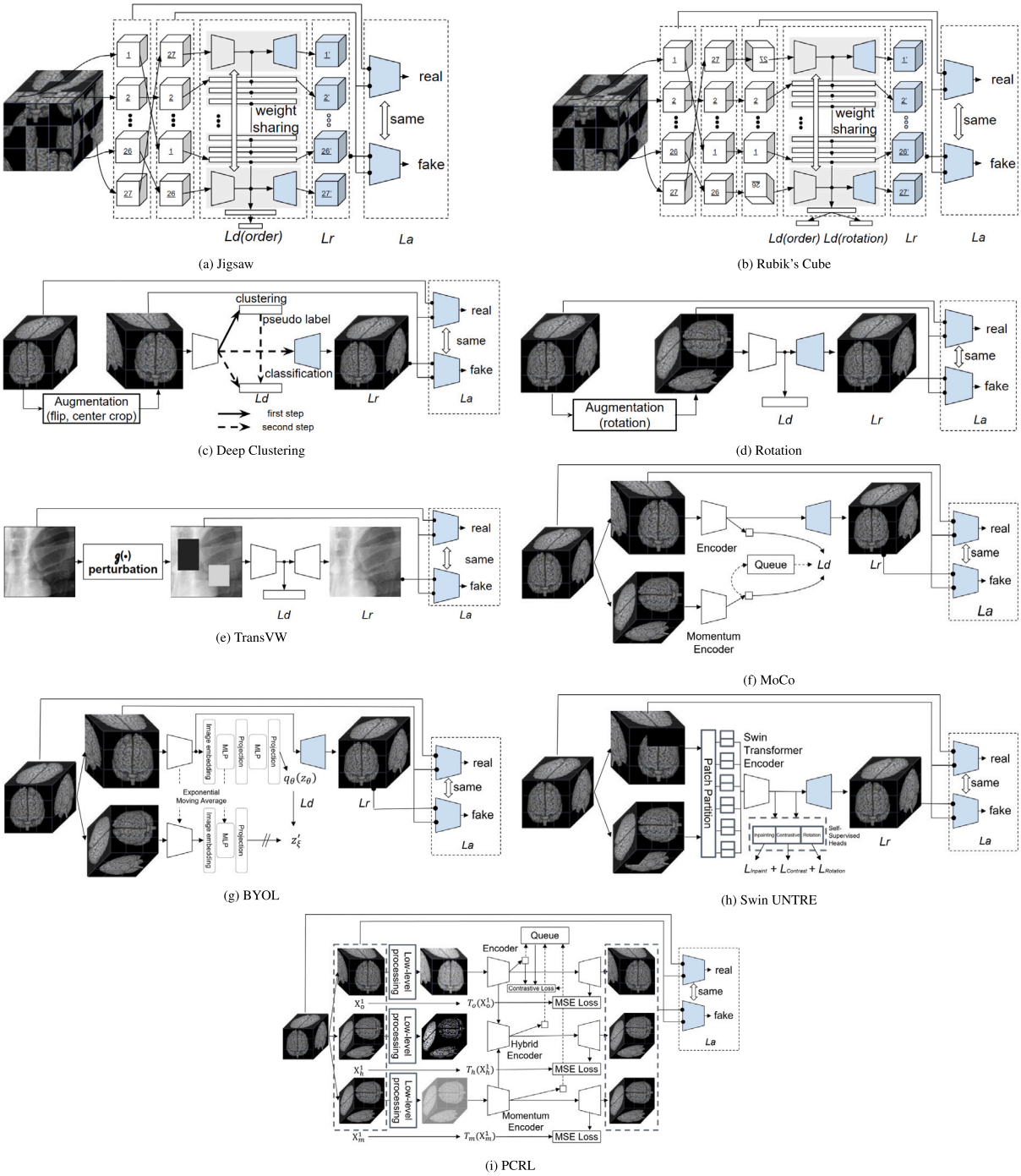


Fig. 2. Redesigning nine prominent SSL methods: (a) Jigsaw, (b) Rubik's Cube, (c) Deep Clustering, (d) Rotation, (e) TransVW, (f) MoCo, (g) BYOL, (h) Swin UNETR, and (i) PCRL in a United framework. The original Jigsaw (Noroozi and Favaro, 2016), Deep Clustering (Caron et al., 2018), Rotation (Gidaris et al., 2018), MoCo (He et al., 2020), and BYOL (Grill et al., 2020) were proposed for 2D image analysis employing discriminative learning alone and provided only pretrained encoders; therefore, in our United framework (a, c, d, f, g), these methods have been augmented with two new components (in light blue) for restorative learning and adversarial learning and re-implemented in 3D. The code for the original Rubik's Cube (Zhuang et al., 2019) is not released and thus reimplemented and augmented with new learning ingredients in light blue (b). The original TransVW (Haghighi et al., 2021), Swin UNETR (Hatamizadeh et al., 2021), and PCRL (Zhou et al., 2021a) are supplemented with adversarial learning (c, h, i). Following our redesign, all nine methods provide all three learned components: discriminative encoders, restorative decoders, and adversary encoders, which are transferable to target classification and segmentation tasks.

PCRL, and Swin UNTRE; and we augmented each with the missing components under our United framework (Fig. 2). A United model (Fig. 1) is a skip-connected encoder–decoder associated with an adversary encoder. With our redesign, for the first time, all nine methods have all three SSL components for 3D medical image analysis.

2.1. Jigsaw

Jigsaw self-supervised learning is a popular technique for training deep neural networks without the need for labeled data. Our 3D Jigsaw approach builds upon the original idea proposed for 2D by Noroozi and

Favaro (2016), extending their notion into 3D as shown in Fig. 2(a). Our 3D jigsaw first divides an input image into a $3 \times 3 \times 3$ grid of 3D patches and shuffles them according to a predefined permutation. To reduce the number of classes, we selected 1000 permutations from all possible combinations using the Hamming Distance-based algorithm (Noroozi and Favaro, 2016). Each permutation is treated as a class, and the Jigsaw puzzle is reformulated as a classification task where the model is trained to recognize the permutation ID.

2.2. Rubik's cube

Similar to the Jigsaw Puzzle pretext task, Rubik's Cube predicts the relative position of sub-cubes in pretext training (Zhuang et al., 2019). It can be seen as the 3D extension of the jigsaw puzzle and naturally takes advantage of volumetric medical image data. Moreover, it is a multitask system that not only predicts the relative position of the sub-cubes but also judges whether each cube has been rotated. This method is a discriminative approach as both pretext tasks are classification problems.

2.3. Deep clustering

Deep clustering extends traditional clustering methods by applying them within neural networks. This method simultaneously learns the parameters of the neural network and the cluster assignment of the extracted features (Caron et al., 2018). It can be viewed as a discriminative method as it learns the parameters through classification tasks. We applied this method to the medical domain for 3D applications by altering the Convolutional Neural Network (CNN) architecture as illustrated in Fig. 2(c).

2.4. Rotation

The rotation-based self-supervised learning method was first introduced by Gidaris et al. (2018). The idea behind this method is to teach a CNN to recognize the rotation angle of an image without the need of human supervision. This is done by defining four possible rotation angles (0, 90, 180, and 270 degrees) and asking the network to predict by which angle the image has been rotated. Building on this concept, (Taleb et al., 2020) proposed a 3D implementation of the rotation-based method. In our work, we adopt their implementation and add restorative and adversarial learning to fit the rotation-based method into our framework.

2.5. TransVW

TransVW is an innovative framework for self-supervised learning that leverages self-discovered visual words as the supervision signal to train a CNN using an encoder–decoder architecture with skip connections and a classification head (Haghighi et al., 2021). The self-discovered visual words are used as the supervision signal. Then, through self-classification, the model is trained to classify each of the visual words. TransVW is very similar to deep clustering, but rather than using the entire image to form clusters, the self-discovering process only considers the patches extracted from the same coordinate across the similar images.

2.6. MoCo

MoCo (He et al., 2020) is an unsupervised visual representation learning technique that makes use of contrastive loss. It includes two encoders, the standard encoder and the momentum encoder. The momentum encoder computes mini-batches and stores them in a queue. The encoders then take the same image with different augmentations and compute the similarity between this encoding and the ones in the queue. The standard encoder is updated using backpropagation, while the momentum encoder is updated through a linear interpolation of the earlier standard encoders.

The training object is formulated using the InfoNCE loss function, which maximizes the similarity between the positive pair and minimizes the similarity between the negative pairs:

$$\mathcal{L}_q = -log \frac{exp(q \cdot k_+/\tau)}{\sum_K^{i=0} exp(q \cdot k_i/\tau)} \tag{1}$$

2.7. BYOL

BYOL (Grill et al., 2020) utilizes a pair of neural networks known as the online and target networks, which collaborate and mutually enhance their learning processes. The online network is trained to predict the target network representation of an image from an augmented view, with the input image presented under a different augmentation. Simultaneously, the target network undergoes updates based on a gradual average of the online network. Notably, BYOL diverges from conventional training methods by not requiring negative samples and abstaining from contrastive loss during its training process.

For a given input image x, BYOL generates two augmented views $v \triangleq t(x)$ and $v' \triangleq t'(x)$. From the initial augmented view v, the online network produces a representation $y_\theta \triangleq f_\theta(v)$ and a corresponding projection $z_\theta \triangleq g_\theta(y)$. Simultaneously, the target network generates $y'_\xi \triangleq f_\xi(v')$ and the associated target projection $z'_\xi \triangleq g_\xi(y')$. The loss is computed using the mean squared error between these two projections:

$$\mathcal{L}_{\theta,\xi} \triangleq \|\bar{q}_\theta(z_\theta) - \bar{z'_\xi}\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2} \tag{2}$$

2.8. PCRL

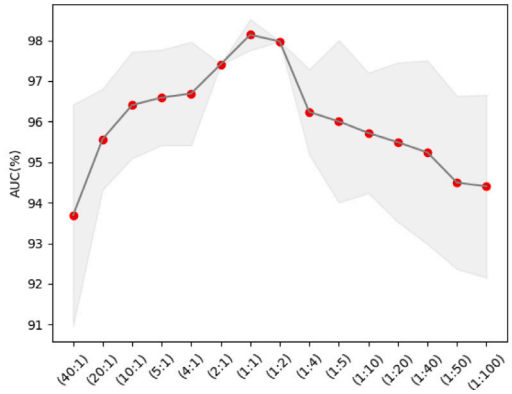
PCRL (Zhou et al., 2021a) combines contrastive and generative self-supervised methods to address the challenge of preserving comprehensive contextual cues in medical images. An innovative aspect involves a generative pretext task that recovers a transformed input using a designated indicator vector, promoting the encoding of richer information. Additionally, a mix-up strategy is employed to diversify image restoration.

2.9. Swin UNETR

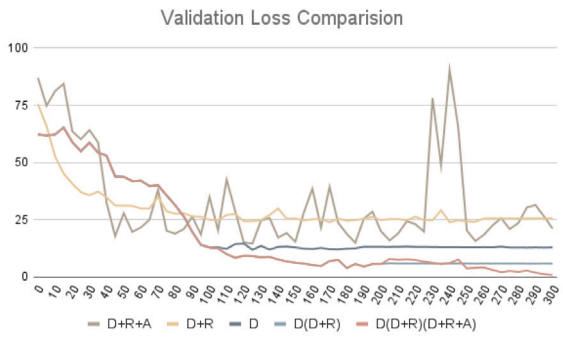
Swin UNETR (Tang et al., 2022) employs a Swin Transformer encoder for processing 3D input patches in pretext tasks. The transformer is pre-trained using self-supervised tasks like image inpainting, 3D rotation prediction, and contrastive learning, utilizing randomly cropped sub-volumes with stochastic data augmentations. The Swin Transformer extracts features at four resolutions via shifted windows for self-attention, connecting to a CNN-based decoder with skip connections at each resolution. This approach efficiently captures global and local information across layers, ensuring scalability for large-scale training.

2.10. Stepwise incremental pretraining

We *incrementally* train United models component-by-component in a *stepwise* manner, yielding three learned transferable components: discriminative encoders, restorative decoders, and adversarial encoders. The pretrained discriminative encoder can be fine-tuned for target classification tasks; the pretrained discriminative encoder and restorative decoder, forming a skip-connected encoder–decoder network (i.e., U-Net (Ronneberger et al., 2015; Siddique et al., 2020)), can be fine-tuned for target segmentation tasks.



(a) Rubik's Cube performance on the NCC task when varying the weights between the discriminative and reconstructive components. The model yields the best performance when $\lambda_d = \lambda_r = 1$.



(b) Validation performance differences between strategy D, D+R, D+R+A, D(D+R), and D(D+R)(D+R+A) for Jigsaw at every 10 epochs. It is apparent that joint training with all three components (i.e., strategy D+R+A) is not stable. The model performs much better in a stepwise incremental manner when we compare D(D+R) with D+R and D(D+R)(D+R+A) with D+R+A

Fig. 3. We determine the best model by varying the weights of the components (a) and adjusting the training strategy (b).

Table 1
When training a United model continually component-by-component, our stepwise incremental pretraining may choose to train the components in different sequences, leading to various pretraining strategies. These strategies can be identified as three types: starting training from the discriminative methods (SDM), start training from the reconstructive methods (SRM), and start training with combined methods (SCM). This table lists the pretraining strategies and their corresponding categories according to all valid component sequences and associates each strategy with its resultant components. For generality, we consider the random initialization as a pretraining strategy, which "generates" randomly-initialized discriminative encoders $\mathcal{E}_\emptyset$, restorative decoders $\mathcal{D}_\emptyset$, and adversary encoders $\mathcal{A}_\emptyset$. For completeness, we list all components with subscripts indicating their pretraining strategies; for those components that cannot be trained by a particular strategy, we indicate them explicitly with subscript $\emptyset$. We evaluate these pretraining strategies in Table 7.

Type	Pretraining strategy	Resultant components
$\emptyset$	Random	$\mathcal{E}_\emptyset$, $\mathcal{D}_\emptyset$, $\mathcal{A}_\emptyset$
SDM	D	$\mathcal{E}_\mathcal{D}$, $\mathcal{D}_\emptyset$, $\mathcal{A}_\emptyset$
	D(D+R)	$\mathcal{E}_{\mathcal{D}(\mathcal{D}+\mathcal{R})}$, $\mathcal{D}_{(\mathcal{D}+\mathcal{R})}$, $\mathcal{A}_\emptyset$
	D(D+R)(D+R+A)	$\mathcal{E}_{\mathcal{D}(\mathcal{D}+\mathcal{R})(\mathcal{D}+\mathcal{R}+\mathcal{A})}$, $\mathcal{D}_{(\mathcal{D}+\mathcal{R})(\mathcal{D}+\mathcal{R}+\mathcal{A})}$, $\mathcal{A}_{(\mathcal{D}+\mathcal{R}+\mathcal{A})}$
SRM	R	$\mathcal{E}_\mathcal{R}$, $\mathcal{D}_\mathcal{R}$, $\mathcal{A}_\emptyset$
	R(R+D)	$\mathcal{E}_{\mathcal{R}(\mathcal{R}+\mathcal{D})}$, $\mathcal{D}_{\mathcal{R}(\mathcal{R}+\mathcal{D})}$, $\mathcal{A}_\emptyset$
	R(R+D)(R+D+A)	$\mathcal{E}_{\mathcal{R}(\mathcal{R}+\mathcal{D})(\mathcal{R}+\mathcal{D}+\mathcal{A})}$, $\mathcal{D}_{\mathcal{R}(\mathcal{R}+\mathcal{D})(\mathcal{R}+\mathcal{D}+\mathcal{A})}$, $\mathcal{A}'_{(\mathcal{R}+\mathcal{D}+\mathcal{A})}$
	R(R+A)	$\mathcal{E}_{\mathcal{R}(\mathcal{R}+\mathcal{A})}$, $\mathcal{D}_{\mathcal{R}(\mathcal{R}+\mathcal{A})}$, $\mathcal{A}_{(\mathcal{R}+\mathcal{A})}$
	R(R+A)(R+A+D)	$\mathcal{E}_{\mathcal{R}(\mathcal{R}+\mathcal{A})(\mathcal{R}+\mathcal{A}+\mathcal{D})}$, $\mathcal{D}_{\mathcal{R}(\mathcal{R}+\mathcal{A})(\mathcal{R}+\mathcal{A}+\mathcal{D})}$, $\mathcal{A}_{(\mathcal{R}+\mathcal{A})(\mathcal{R}+\mathcal{A}+\mathcal{D})}$
SCM	(D+R)	$\mathcal{E}_{(\mathcal{D}+\mathcal{R})}$, $\mathcal{D}_{(\mathcal{D}+\mathcal{R})}$, $\mathcal{A}_\emptyset$
	(D+R)(D+R+A)	$\mathcal{E}_{(\mathcal{D}+\mathcal{R})(\mathcal{D}+\mathcal{R}+\mathcal{A})}$, $\mathcal{D}_{(\mathcal{D}+\mathcal{R})(\mathcal{D}+\mathcal{R}+\mathcal{A})}$, $\mathcal{A}''_{(\mathcal{D}+\mathcal{R}+\mathcal{A})}$
	(R+A)	$\mathcal{E}_{(\mathcal{R}+\mathcal{A})}$, $\mathcal{D}_{(\mathcal{R}+\mathcal{A})}$, $\mathcal{A}_{(\mathcal{R}+\mathcal{A})}$
	(R+A)(R+A+D)	$\mathcal{E}_{(\mathcal{R}+\mathcal{A})(\mathcal{R}+\mathcal{A}+\mathcal{D})}$, $\mathcal{D}_{(\mathcal{R}+\mathcal{A})(\mathcal{R}+\mathcal{A}+\mathcal{D})}$, $\mathcal{A}_{(\mathcal{R}+\mathcal{A})(\mathcal{R}+\mathcal{A}+\mathcal{D})}$
	(D+R+A)	$\mathcal{E}_{(\mathcal{D}+\mathcal{R}+\mathcal{A})}$, $\mathcal{D}_{(\mathcal{D}+\mathcal{R}+\mathcal{A})}$, $\mathcal{A}'''_{(\mathcal{D}+\mathcal{R}+\mathcal{A})}$

Discriminative learning trains a discriminative encoder D_θ, where θ represents the model parameters, to predict target label $y \in Y$ from input $x \in X$ by minimizing a loss function for $\forall x \in X$ defined as

$$\mathcal{L}_d = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \ln(p_{nk}) \tag{3}$$

where N is the number of samples, K is the number of classes, and p_{nk} is the probability predicted by D_θ for x_n belonging to Class k; that is, $p_n = D_\theta(x_n)$ is the probability distribution predicted by D_θ for x_n over all classes. In SSL, the labels are automatically obtained based on the properties of the input data, involving no manual annotation. All nine SSL methods in this work have a discriminative component formulated as a classification task, while other discriminative losses can be used, such as contrastive losses in MoCo-v2 (Chen et al., 2020), Barlow Twins (Zbontar et al., 2021), and SimSiam (Chen and He, 2021).

Restorative learning trains an encoder–decoder $(D_\theta, R_{\theta'})$ to reconstruct an original image x from its distorted version $\mathcal{T}(x)$, where $\mathcal{T}$ is a distortion function, by minimizing pixel-level reconstruction error:

$$\mathcal{L}_r = \mathbb{E}_x\ L_2(x, R_{\theta'}(D_\theta(\mathcal{T}(x)))) \tag{4}$$

where $L_2(u, v)$ is the sum of squared pixel-by-pixel differences between u and v.

Adversarial learning trains an additional adversary encoder, $A_{\theta''}$, to help the encoder–decoder $(D_\theta, R_{\theta'})$ reconstruct more realistic medical images and, in turn, strengthen representation learning. The adversary encoder learns to distinguish the fake image pair $(R_{\theta'}(D_\theta(\mathcal{T}(x))), \mathcal{T}(x))$ from the real pair $(x, \mathcal{T}(x))$ via an adversarial loss:

$$\mathcal{L}_a = E_{x,\mathcal{T}(x)} log A_{\theta''}(\mathcal{T}(x), x) + E_x log(1 - A_{\theta''}(\mathcal{T}(x), R_{\theta'}(D_\theta(\mathcal{T}(x))))) \tag{5}$$

The final objective combines all losses:

$$\mathcal{L} = \lambda_d \mathcal{L}_d + \lambda_r \mathcal{L}_r + \lambda_a \mathcal{L}_a \tag{6}$$

where λ_d, λ_r, and λ_a controls the importance of each learning ingredient. A grid-search hyper-parameter optimization was performed which estimated the optimal values of $\lambda_d = 1$, $\lambda_r = 1$, and $\lambda_a = 10$.

Stepwise incremental pretraining trains a United model continually component-by-component because the model's complexity makes it difficult to train the whole model in an end-to-end fashion (i.e., all three components together directly from scratch), a strategy called D+R+A. As depicted in Fig. 3(b), the validation performance of Strategy D+R+A fluctuates significantly during the training process. Strategy D+R+A is always outperformed by, for example, Strategy D(D+R)(D+R+A), as illustrated in Fig. 1 and Strategy D(D+R)(D+R+A) provides the most reliable performance across most target tasks evaluated in this work (see Table 2). When training a United model continually component-by-component, our stepwise incremental pretraining may follow different component sequences, leading to various pretraining strategies as summarized in Table 1. We compare these pretraining strategies in Section 5.1 and in Table 7.

Table 2
Strategy D(D+R)(D+R+A) always outperforms Strategy D+R+A on all six target tasks. We include the mean and standard deviation from ten runs and an independent two-sample t-test between the two strategies. The text is bolded when they are significantly different at p = 0.05 level.

Method	Approach	Pretrained component utilized for classification	ECC	NCC	VCC
Jigsaw	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$	84.12 ± 1.38	97.24 ± 0.73	91.62 ± 0.84
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$	84.89 ± 1.05	97.86 ± 1.54	**92.49 ± 0.51**
Rubik's Cube	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$	84.36 ± 1.17	98.21 ± 0.88	91.8 ± 1.32
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$	**85.64 ± 0.87**	**99.17 ± 0.79**	**92.66 ± 0.57**
TransVW	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$	85.84 ± 1.84	97.63 ± 0.52	92.03 ± 0.96
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$	**86.91 ± 3.27**	**98.14 ± 0.44**	**92.95 ± 1.13**
MoCo	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$	84.84 ± 2.72	98.19 ± 0.41	91.74 ± 2.52
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$	**86.68 ± 1.69**	**98.63 ± 0.44**	**93.87 ± 0.83**
Method	Approach	Pretrained components utilized for segmentation	BMS	NCS	LCS
Jigsaw	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R+\mathcal{A})}$	64.98 ± 0.68	74.32 ± 1.54	83.54 ± 0.95
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$, $\mathcal{D}_{((D+R)D+R+\mathcal{A})}$	**66.07 ± 1.33**	74.87 ± 1.17	**84.87 ± 1.67**
Rubik's Cube	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R+\mathcal{A})}$	65.13 ± 1.34	75.18 ± 1.32	84.12 ± 1.19
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R)(D+R+\mathcal{A})}$	**66.88 ± 1.72**	76.07 ± 1.23	**85.18 ± 0.99**
TransVW	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R+\mathcal{A})}$	66.81 ± 1.06	76.32 ± 1.25	85.16 ± 0.67
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R)(D+R+\mathcal{A})}$	**69.57 ± 1.13**	**77.51 ± 1.36**	**86.85 ± 0.81**
MoCo	D+R+A	$\mathcal{E}_{(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R+\mathcal{A})}$	67.04 ± 0.76	80.41 ± 0.36	86.09 ± 1.40
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$, $\mathcal{D}_{(D+R)(D+R+\mathcal{A})}$	**69.53 ± 0.81**	80.47 ± 0.61	86.71 ± 0.94

3. Experimental setup

Model: We utilize the U-Net model with skip connections (Ronneberger et al., 2015; Siddique et al., 2020) in our study. This model has demonstrated state-of-the-art performance for medical imaging segmentation tasks, and we used its encoder part for classification tasks. For each of the nine methods, we redesigned the model to incorporate all three learning components: discriminative, restorative, and adversarial.

Fine-tuning: For all experiments, we fine-tune the pretrained model end-to-end on the target transfer dataset. The datasets used for pretraining and fine-tuning are introduced below.

Datasets and Metrics: We used 623 CT scans from the LUNA16 (Setio et al., 2017) dataset to pretrain all nine of our models. We adopted the same approach as (Zhou et al., 2021b) and extracted sub-volumes with a size of 64 × 64 × 64 pixels. To assess the usefulness of pretraining the nine models, we tested them on nine 3D medical imaging tasks including BraTS (Menze et al., 2014; Bakas et al., 2018), LUNA16 (Setio et al., 2017), LIDC-IDRI (Armato III et al., 2011), PE-CAD (Tajbakhsh et al., 2015), PE-CAD (VOIR) (Tajbakhsh et al., 2019), and LiTS (Bilic et al., 2019). These tasks are BMS (brain tumor segmentation), NCC (reducing lung nodule false positives), NCS (lung nodule segmentation), ECC (reducing pulmonary embolism false positives), VCC (reducing pulmonary embolism false positives with vessel-oriented image representation), and LCS (liver segmentation). We calculated the efficacy of the pretrained models on the nine target tasks and reported the AUC for classification tasks and IoU for segmentation tasks. All target tasks were executed at least 10 times, and statistical analysis was performed using the independent two-sample t-test.

Brain tumor segmentation (BMS): The dataset, which comes from the BraTS 2018 challenge (Menze et al., 2014; Bakas et al., 2018), includes 285 patients (210 HGG and 75 LGG), each with four rigorously aligned 3D MRI modalities (T1, T1c, T2, and Flair). In our 3-fold cross validation method, 95 patients comprised the test fold while 190 patients comprised the training fold. Three tumor subregions were annotated: the necrotic and non-enhancing tumor core (label 1), the GD-enhancing tumor (label 4), and the peritumoral edema (label 2). Here, the background was annotated (label 0). Finally, Intersection over Union (IoU) was used to assess segmentation performance. We treated those with label 0 as negatives and all other data as positives.

Lung nodule false positive reduction (NCC): The dataset is from LUNA16 (Setio et al., 2017) which consists of 888 CT scans with a slice thickness less than 2.5 mm. With 445, 265, and 178 instances each, the dataset is subdivided into training, validation, and testing sets. The initial data were made available for segmenting lung nodules, but additional annotation was made available for the task of reducing false-positive results. Following prior work (Zhou et al., 2021b; Haghighi et al., 2021), we evaluated the performance using the AUC score for classifying true positives and false positive results.

Lung nodule segmentation (NCS): The dataset is made available by the Lung Image Database Consortium image collection (LIDC-IDRI) (Armato III et al., 2011) with 1088 cases consisting of lung CT scans with masked nodule locations. The training set contains 510 cases, the validation set includes 100 cases, and the testing set includes 480 cases. To train using this dataset, the CT scans are re-sampled to 1-1-1 spacing, and we extract cubes with a size of 64 × 64 × 32. Following previous work (Zhou et al., 2021b; Haghighi et al., 2021), we adopted Intersection over Union (IoU) to evaluate performance.

Pulmonary embolism false positive reduction (ECC): We employed a database that contains 326 emboli from 121 computed tomography pulmonary angiography (CTPA) images. Following the work of Liang and Bi (2007), we used the proprietary algorithm-based PE candidate generator, which yielded a total of 687 true positives and 5568 false positives. The dataset was then split into a training and a testing set. The training set contains 434 true positive PE candidates and 3406 false positive PE candidates. The testing set contains 253 true positive PE candidates and 2162 false positive PE candidates, both at the patient-level. We calculated the candidate level AUC for distinguishing true and false positive results to facilitate an accurate comparison with the previous study.

Pulmonary embolism false positive reduction with vessel-oriented image representation (VCC): In this task, we focus on using vessel-oriented image representation (VOIR) to improve the accuracy of image representations of PE candidates (Tajbakhsh et al., 2019). By aligning the image planes with the vessel longitudinal axis, the VOIR approach maximizes the visualization of pulmonary arterial filling defects and generates more accurate representations of PE candidates. We further extend the VOIR into 3D from Tajbakhsh et al. (2019) and evaluate the performance of all nine methods on the false positive reduction task by calculating the candidate level AUC.

Liver segmentation (LCS): A total of 130 labeled CT scans from the MICCAI (Bilic et al., 2019), LiTS Challenge dataset were divided into subgroups for training (100 patients), validation (15 patients), and testing (15 patients). Two distinct labels, liver and lesion, were provided by the ground truth segmentation. We used IoU to assess segmentation performance in our studies, regarding only the liver as a "positive class", and all other classes as "negative class".

Table 3
Discriminative learning enhances discriminative encoders for classification and segmentation tasks. We report the mean and standard deviation (mean ± s.d.) across ten trials, along with the statistic analysis with and without discriminative pretraining for nine self-supervised learning methods. With discriminative pretraining, the performance gains were observed across target tasks with exceptions in NCS for Jigsaw and Rubik's Cube and in LCS for Rubik's Cube, where the performance of the pretrained model is worse than random initialization due to possible incompatibilities between pretraining and targets.

Method	Pretrained component utilized for classification	ECC	NCC	VCC
Random	$\mathcal{E}_{\emptyset}$	79.99 ± 8.06	94.25 ± 5.07	91.35 ± 1.34
Jigsaw	$\mathcal{E}_D$	81.79 ± 1.04*	95.49 ± 1.24*	91.51 ± 1.09
Rubik's Cube		81.76 ± 1.32*	96.24 ± 1.57*	91.45 ± 1.35
Deep Clustering		84.82 ± 0.62***	97.27 ± 1.43***	91.87 ± 1.3
TransVW		84.25 ± 3.91***	97.49 ± 0.45***	92.04 ± 1.08**
Rotation		82.37 ± 1.64**	96.13 ± 2.41*	91.52 ± 1.72
MoCo		85.53 ± 1.97***	98.42 ± 0.41***	92.18 ± 0.71**
BYOL		86.34 ± 0.63***	98.72 ± 0.44***	92.57 ± 0.64***
PCRL		85.01 ± 0.42***	98.06 ± 0.19***	92.17 ± 1.58**
Swin UNETR		85.54 ± 0.42***	98.41 ± 0.33***	92.86 ± 0.89***
Method	Pretrained components utilized for segmentation	BMS	NCS	LCS
Random	$\mathcal{E}_{\emptyset}$, $D_{\emptyset}$	58.52 ± 2.61	74.05 ± 1.97	77.82 ± 3.87
Jigsaw	$\mathcal{E}_D$, $D_{\emptyset}$	63.33 ± 1.11**	73.38 ± 1.65*	82.04 ± 1.65*
Rubik's Cube		62.75 ± 1.93**	72.87 ± 0.86**	77.42 ± 0.43*
Deep Clustering		65.81 ± 0.73***	74.82 ± 0.47*	82.67 ± 0.69**
TransVW		64.02 ± 0.98**	76.93 ± 0.87***	85.09 ± 2.15***
Rotation		63.98 ± 0.84**	74.24 ± 0.91*	82.44 ± 1.45**
MoCo		69.19 ± 0.64***	80.41 ± 0.36***	86.12 ± 0.99***
BYOL		68.93 ± 0.64***	80.70 ± 0.56***	85.32 ± 0.72***
PCRL		68.66 ± 0.42***	79.77 ± 0.75***	85.51 ± 0.19***
Swin UNETR		68.59 ± 0.22***	80.29 ± 0.31***	85.81 ± 0.34***

* $p < 0.5$.
** $p < 0.1$.
*** $p < 0.05$.

Table 4
Incremental restorative pretraining combined with continual discriminative learning (*i.e.*, Strategy D(D+R)) enhances discriminative encoders for classification tasks. We report the mean and standard deviation (mean ± s.d.) across ten trials, along with the statistic analysis with and without incremental restorative pretraining for nine self-supervised learning methods. With Strategy D(D+R), the performance gains from $\mathcal{E}_{D(D+R)}$ were consistent for all target tasks in comparison with $\mathcal{E}_D$.

Method	Approach	Pretrained component(s) utilized for classification	NCC	ECC	VCC
Random	–	$\mathcal{E}_{\emptyset}$	94.25 ± 5.07	79.99 ± 8.06	91.35 ± 1.34
Jigsaw	D	$\mathcal{E}_D$	95.49 ± 1.24	81.79 ± 1.04	91.51 ± 1.09
	D(D+R)	$\mathcal{E}_{D(D+R)}$	97.29 ± 1.09***	84.39 ± 1.47***	92.3 ± 0.57**
Rubik's Cube	D	$\mathcal{E}_D$	96.24 ± 1.57	81.76 ± 1.32	91.45 ± 1.35
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.14 ± 0.38***	84.14 ± 1.58***	92.39 ± 0.69*
Deep Clustering	D	$\mathcal{E}_D$	97.27 ± 1.43	84.82 ± 0.62	91.87 ± 1.3
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.11 ± 0.55	85.12 ± 1.37	92.14 ± 0.98*
TransVW	D	$\mathcal{E}_D$	97.49 ± 0.45	84.25 ± 3.91	92.04 ± 1.08
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.47 ± 0.22*	87.07 ± 2.83*	92.57 ± 0.76*
Rotation	D	$\mathcal{E}_D$	96.13 ± 2.41	82.37 ± 1.64	91.08 ± 1.41
	D(D+R)	$\mathcal{E}_{D(D+R)}$	97.17 ± 0.81	83.57 ± 1.21*	91.25 ± 0.48
MoCo	D	$\mathcal{E}_D$	98.42 ± 0.41	85.53 ± 1.97	92.18 ± 0.71
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.65 ± 0.13*	86.93 ± 1.25*	93.68 ± 0.78
BYOL	D	$\mathcal{E}_D$	98.72 ± 0.44	86.34 ± 0.63	92.57 ± 0.64
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.84 ± 0.21	86.50 ± 0.31	93.46 ± 0.42
PCRL	D	$\mathcal{E}_D$	98.06 ± 0.19	85.01 ± 0.42	92.17 ± 1.58
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.39 ± 0.23	85.85 ± 0.48*	92.58 ± 1.14
Swin UNETR	D	$\mathcal{E}_D$	98.41 ± 0.33	85.54 ± 0.42	92.86 ± 0.89
	D(D+R)	$\mathcal{E}_{D(D+R)}$	98.67 ± 0.20	86.38 ± 0.35*	93.58 ± 0.77

* $p \leq 0.05$.
** $p \leq 0.01$.
*** $p \leq 0.001$.

4. Experiments and results

In this section, we investigate the importance of the incremental pretraining strategy in the United framework. Further, we discuss how to utilize each component in the United framework for downstream tasks.

4.1. Discriminative encoders ($\mathcal{E}_D$) are useful for both classification and segmentation tasks

We train the discriminative encoders using nine SSL methods and apply them to six target tasks. The discriminative learning significantly enhances encoders in both classification and segmentation tasks, as

Table 5
Incremental restorative pretraining combined with continual discriminative learning (*i.e.*, Strategy D(D+R)) directly boosts target segmentation tasks. Statistic analysis was conducted between using incremental restorative pretrained decoder ($D_{(D+R)}$) and using random decoder ($D_{\emptyset}$). With Strategy D(D+R), the segmentation performance gains from ($\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$) were consistent for all target tasks in comparison with ($\mathcal{E}_{\emptyset}$, $D_{\emptyset}$) and ($\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$). We use ($\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$) to indicate that pretrained $\mathcal{E}_{D(D+R)}$ is attached with a randomly-initialized decoder $D_{\emptyset}$ to form a U-Net for segmentation without using pretrained $D_{(D+R)}$, even though we have it, to highlight the capability of $D_{(D+R)}$ in boosting target segmentation performance.

Method	Approach	Pretrained components utilized for segmentation	NCS	LCS	BMS
Random	–	$\mathcal{E}_{\emptyset}$, $D_{\emptyset}$	74.05 ± 1.97	77.82 ± 3.87	58.52 ± 2.61
Jigsaw	D(D+R)	$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	73.58 ± 1.26	83.04 ± 1.21	64.17 ± 0.62
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	74.53 ± 1.13*	84.17 ± 1.48***	65.33 ± 1.31***
Rubik's Cube		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	74.33 ± 1.83	84.21 ± 0.24	64.91 ± 0.76
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	75.66 ± 0.74***	85.02 ± 1.08***	65.83 ± 1.16***
Deep Clustering		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	75.01 ± 0.69	83.75 ± 0.9	66.14 ± 0.87
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	75.91 ± 1.12***	84.63 ± 0.63***	66.73 ± 0.51***
TransVW		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	77.09 ± 1.52	85.63 ± 0.96	67.52 ± 0.87
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	77.33 ± 0.52	86.53 ± 1.31*	68.82 ± 0.38***
Rotation		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	74.65 ± 1.26	83.24 ± 2.21	64.54 ± 1.36
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	74.86 ± 0.58*	84.65 ± 1.01***	65.44 ± 0.67***
MoCo		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	80.63 ± 1.01	86.3 ± 0.51	69.47 ± 0.58
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	80.74 ± 0.36*	86.72 ± 0.60*	69.66 ± 0.41*
BYOL		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	80.75 ± 1.24	85.72 ± 0.71	69.08 ± 0.93
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	80.80 ± 0.73*	86.14 ± 0.37**	69.11 ± 0.47
PCRL		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	80.53 ± 0.85	85.83 ± 0.62	69.02 ± 0.36
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	80.64 ± 0.41	86.04 ± 0.53*	69.04 ± 0.27
Swin UNTRE		$\mathcal{E}_{D(D+R)}$, $D_{\emptyset}$	80.61 ± 1.42	86.53 ± 0.79	69.30 ± 0.77
		$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	80.69 ± 0.18	86.75 ± 0.28	69.52 ± 0.19*

* $p < 0.5$.
** $p < 0.1$.
*** $p < 0.05$.

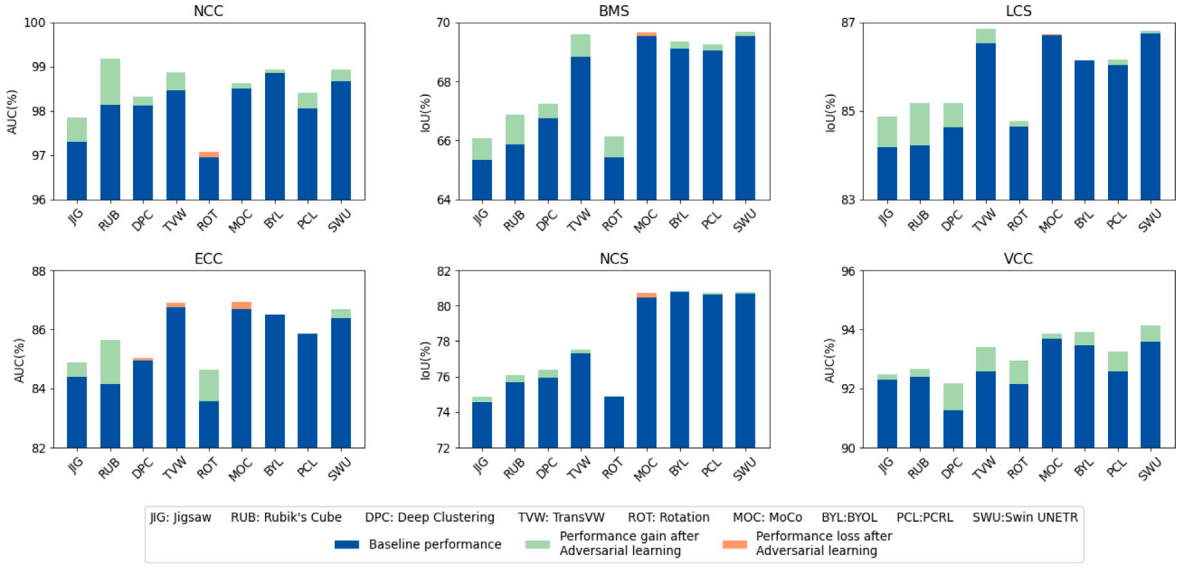


Fig. 4. Incremental adversarial training in Strategy D(D+)(D+R+A) strengthens learned representation. Target task performance is generally increased (red) following the adversarial training. Although some target tasks show a decrease (pink), these reductions are not statistically significant, according to the t-test.

shown in Table 3. Specifically, compared with the random initialization, the Deep Clustering method improved NCC, ECC, NCS, LCS, BMS, and VCC by AUC scores of 3.0%, 4.8%, 0.8%, 4.8%, 7.3%, and 0.5%, respectively. Similarly, TransVW improves the target tasks by 3.2%, 4.3%, 2.9%, 7.2%, 5.5%, and 0.7%, Rotation by 1.9%, 2.4%, 0.2%, 4.6%, 5.5%, and 0.2%, MoCo by 4.2%, 5.5%, 6.4%, 8.3%, 10.6%, and 0.8%, BYOL by 0.1%, 0.2%, 0.1%, 0.8%, 0.2%, and 0.9%, PSL by 0.3%, 0.8%, 0.9%, 0.5%, 0.4%, and 0.4%, and SWU by 0.3%, 0.87%, 0.4%, 0.9%, 0.9%, 0.7%. The Jigsaw method improved NCC, ECC, LCS, BMS, and VCC by AUC scores of 1.3%, 1.8%, 4.2%, 3.8%, and 0.2%, respectively. The Rubik's Cube method improved in NCC, ECC, BMS, and VCC by AUC scores of 2.0%, 1.8%, 4.2%, and 0.1%, respectively.

4.2. Incremental restorative pretraining combined with continual discriminative learning (i.e., strategy D(D+R)) further enhances discriminative encoders for classification tasks

After pretraining discriminative encoders, we append restorative decoders to the end of the encoders and continue to pretrain them together. The incremental restorative learning significantly enhances

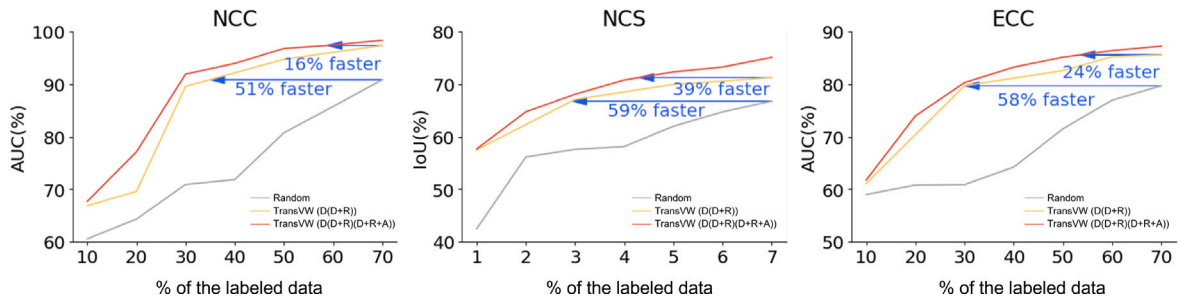


Fig. 5. Stepwise incremental pretraining D(D+R)(D+R+A) helps reduce annotation costs. As an example, for target tasks of NCC, NCS, and ECC, incremental pretrained TransVW with Strategy D(D+R)(D+R+A) reduces the annotation cost by 28%, 43%, and 26%, respectively, in comparison with Strategy D(D+R), and by 57%, 61%, and 66%, respectively, comparison with training from scratch.

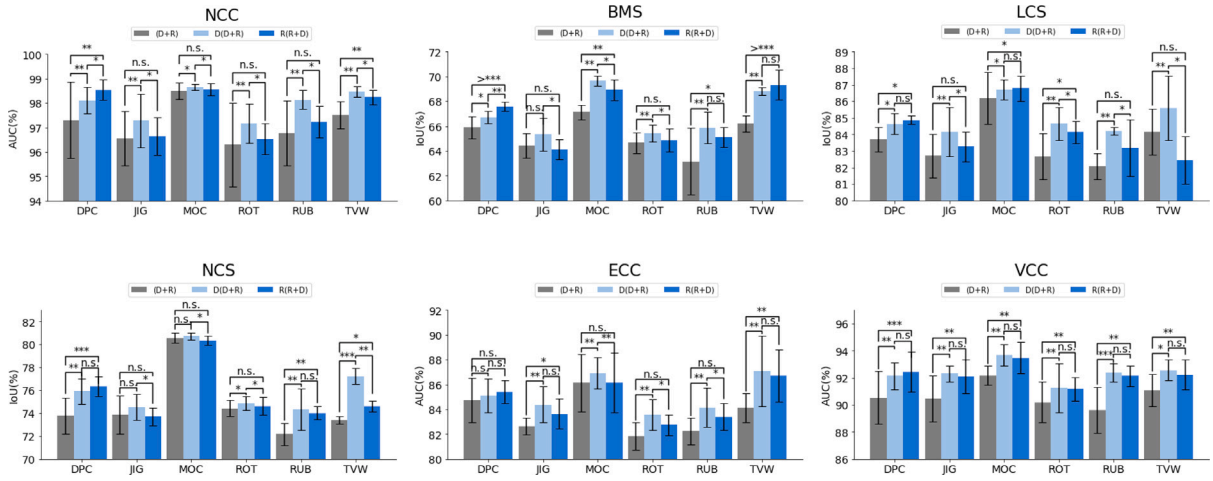


Fig. 6. We tested different variants of the stepwise incremental pretraining scheme R(R+D) and D(D+R) with components D and R. Compared to the end-to-end training scheme (D+R), R(R+D) increased the target task performance the majority of the time. The only two performance decreases are for Jigsaw on BMS and NCS but they were not significant according to the t-test. D(D+R) always improved performance on all five target tasks compared to (D+R). For five out of six methods, D(D+R) also had better performance on all five target tasks compared to R(R+D). The only exception is for the Deep Clustering where R(R+D) always performs better than D(D+R). We believe that this is because for Deep Clustering, training the reconstruction (R) first helps to initialize the clusters, yielding better overall performance.

encoders in classification tasks, as shown in Table 4. Specifically, compared with the original methods, the incremental restorative learning improves Jigsaw by AUC scores of 1.9%, 2.6%, and 0.8% in NCC, ECC, and VCC; similarly, it improves Rubik's Cube by 1.9%, 2.4%, and 0.9%, Deep Clustering by 0.9%, 0.3%, and 0.3%, TransVW by 1.0%, 2.9%, and 0.5%, Rotation by 1.0%, 1.2%, and 0.2%, MoCo by 0.2%, 1.4%, and 1.5%, BYOL by 0.1%, 0.2%, and 0.9%, PCRL by 0.3%, 0.8%, and 0.4%, and Swin UNTRE by 0.3%, 0.8%, and 0.7%. The discriminative encoders are enhanced because they learn global features along with fine-grained features through incremental restorative learning.

4.3. Incremental restorative pretraining combined with continual discriminative learning (i.e., strategy D(D+R)) directly boosts target segmentation tasks

Most state-of-the-art segmentation methods do not pretrain their decoders, but instead initialize them at random (He et al., 2020; Chen et al., 2020). Table 5 shows that the random decoders are suboptimal, while incremental pretrained restorative decoders can significantly improve target segmentation tasks. Specifically, compared with the D methods, the incremental pretrained restorative decoder improves Jigsaw by 1.2%, 2.1% and 2.0% IoU improvement in NCS, LCS and BMS, respectively. Similarly, it improves Rubik's Cube by 2.8%, 7.6%, and 3.1%; Deep Clustering by 1.1%, 2.0%, and 0.9%; TransVW by 0.4%, 1.4%, and 4.8%; Rotation by 0.6%, 2.2% and 1.5%; MoCo by 0.1%, 0.4%, and 0.2%, BYOL by 0.2%, 0.4%, and 0.1%, PCRL by 0.1%, 0.2%, and 0.1%, and Swin UNTRE by 0.1%, 0.2%, and 0.2%. The consistent performance gains indicate that a wide variety of target segmentation tasks can benefit from our incremental pretrained restorative decoders.

4.4. Strategy D(D+R)(D+R+A) strengthens representation learning and reduces annotation costs

Quantitative measurements shown in Table 6 reveal that adversarial training can generate sharper and more realistic images in the restoration proxy task. More importantly, we found that adversarial training also makes a significant contribution to pretraining. First, as shown in Fig. 4, adding adversarial training can benefit most target tasks, particularly segmentation tasks. The incremental adversarial pretraining improves Jigsaw by AUC scores of 0.3%, 0.7%, and 0.7% in NCS, LCS, and BMS, respectively. Similarly, it improves Rubik's Cube by 0.4%, 1.0%, and 1.0%; Deep Clustering by 0.5%, 0.5%, and 0.5%; TransVW by 0.2%, 0.3%, and 0.8%; Rotation by 0.1%, 0.1%, and 0.7%; BYOL by 0.1%, 0.1%, and 0.1%, PCRL by 0.1%, 0.1%, and 0.1%, and Swin UNTRE 0.3%, 0.2%, and 0.2%. Additionally, incremental adversarial pretraining improves performance on small data regimes. Fig. 5 shows that incremental adversarial pretrained TransVW (Haghighi et al., 2021) can reduce the annotation cost by 28%, 43%, and 26% on NCC, NCS, and ECC, respectively, compared with TransVW (Haghighi et al., 2021).

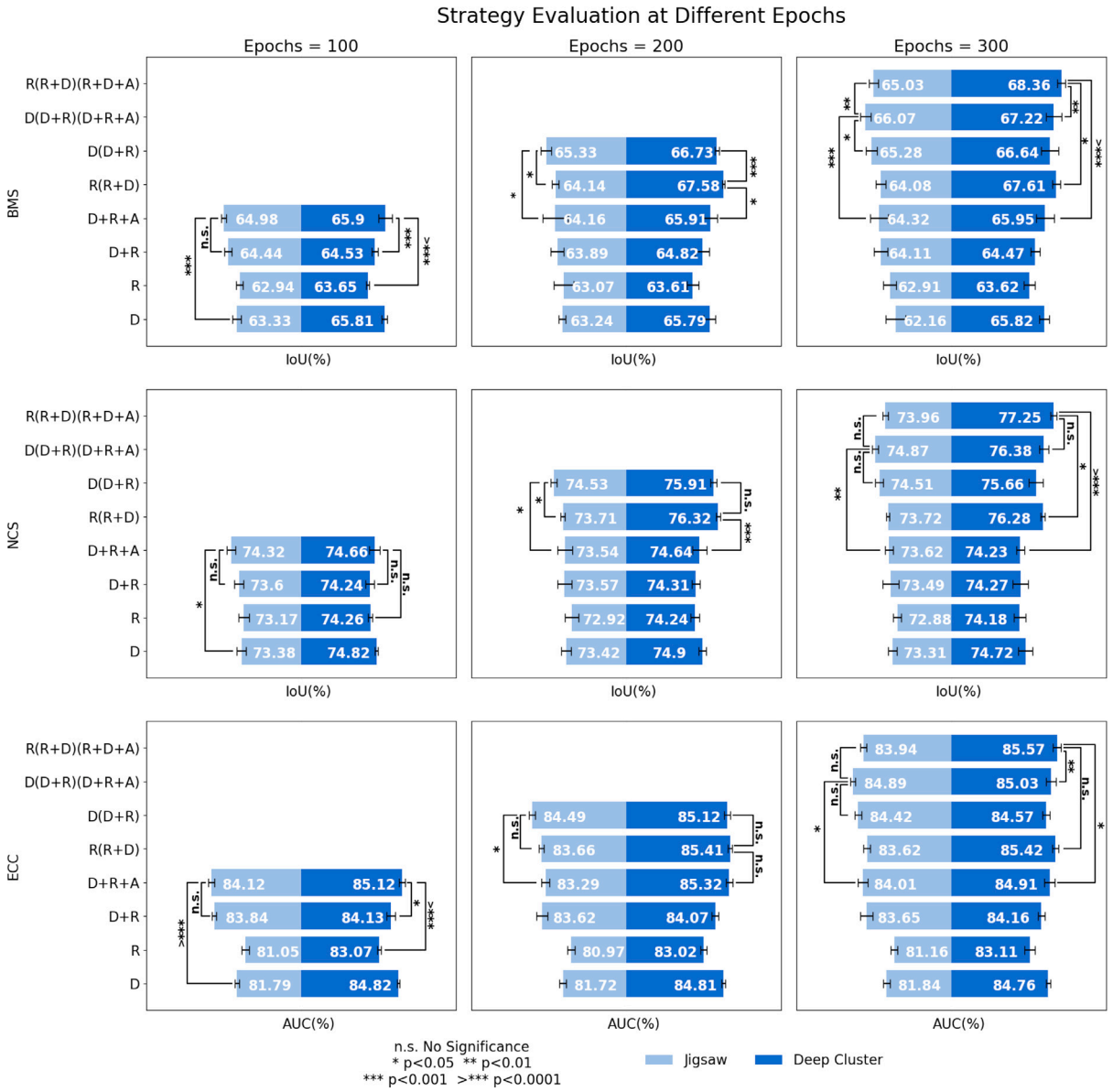


Fig. 7. Target task performance (BMS, NCS, and ECC) for Jigsaw and Deep Clustering trained at 100, 200, and 300 epochs. For each method, we employ eight strategies to train the model, including D, D+R, D+R+A, D(D+R), R(R+D), D(D+R)(D+R+A), and R(R+D)(R+D+A), which are the most representative. In stepwise incremental pretraining, it is typical for one step to be taken for 100 epochs. Therefore, for example, at 100 epochs, the model with the strategy D(D+R) does not exist. For the sake of fair comparison, we prolong the training of, for example, strategy D, to 200 and 300 epochs to compare it with other strategies that are trained with the same number of epochs. Over additional epochs, incrementally pretrained models consistently outperform jointly trained ones. For example, at 200 epochs, D(D+R) and R(R+D) surpass D+R+A, while at 300 epochs, D(D+R)(D+R+A) and R(R+D)(R+D+A) yield the best performance.

5. Discussion

5.1. Comparing different incremental pretraining strategies in the unified framework for downstream tasks

The self-supervised methods we selected are primarily discriminative/contrastive methods, with reconstructive and adversarial components being universal across all methods. It is possible to vary the reconstructive and adversarial components while maintaining the same discriminative/contrastive component across all the methods. However, this would introduce an exponentially larger number of combinations, which is beyond the scope of this work. When the only variable becomes the discriminative component, we further identify two types of discriminative methods: clustering or non-clustering. We then test the performance of each methods through different training strategies.

The training strategies can also be identified as three types: starting training from the discriminative methods (SDM), start training from the reconstructive methods (SRM), and start training with combined methods (SCM). The SDM strategy includes D, D(D+R), and D(D+R)(D+R+A). The SRM strategy includes R, R(R+D), R(R+A), R(R+D)(R+D+A), and R(R+A)(R+A+D). The SCM strategy includes (D+R), (D+R)(D+R+A), (R+A)(R+A+D), and (D+R+A).

The combined use of discriminative and restorative methods (strategy D+R) consistently outperforms the individual use of either method (strategies D or R), as evident in Table 7. Furthermore, the models' performances are further enhanced with the pretraining of one of the methods (D or R). As shown in Fig. 6, D(D+R) is always better than (D+R) across all target tasks with all nine methods and generally better than R(R+D) except for Deep Clustering and TransVW. At last, we perform pretraining with all three components (D, R, and A) and observe that the stepwise incremental pretraining strategy consistently outperforms the combined training strategy, with the

Table 6
The final stepwise incremental pretraining (Step D(D+R)(D+R+A)) generates sharper and more realistic images for restoration tasks. After further adversarial training, the MSE and FID scores for each of the nine approaches all declined, suggesting that the produced images' distribution had moved closer to the original one. The MS-SSIM score increased after the adversarial training, indicating the generated images were structurally similar to the original one.

Method	Adv.	Pretrained components utilized	MSE (↓)	FID (↓)	MS-SSIM (↑)
Jigsaw	✗	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	0.0168 ± 0.0024	338.245	0.8335 ± 0.0024
	✓	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	0.0143 ± 0.0017	317.354	0.8724 ± 0.0012
Rubik's Cube	✗	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	0.0139 ± 0.0011	314.323	0.8856 ± 0.0015
	✓	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	0.0115 ± 0.0005	257.698	0.9127 ± 0.0007
Deep Clustering	✗	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	0.0123 ± 0.0019	295.645	0.8973 ± 0.0021
	✓	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	0.0108 ± 0.0012	244.742	0.9268 ± 0.0018
TransVW	✗	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	0.0289 ± 0.0027	427.562	0.7383 ± 0.0032
	✓	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	0.0109 ± 0.0015	251.325	0.9088 ± 0.0015
Rotation	✗	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	0.0184 ± 0.0052	356.32	0.7914 ± 0.0032
	✓	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	0.0129 ± 0.0021	309.214	0.8932 ± 0.0028
MoCo	✗	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	0.0097 ± 0.0012	221.42	0.9324 ± 0.0014
	✓	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	0.0075 ± 0.0008	204.58	0.9536 ± 0.0037

Table 7
Comparing different training strategies. We report the mean and standard deviation (mean ± s.d.) based on ten trials, along with the statistic analysis between the best (highest mean value) and the worst (lowest mean value) training strategies among similar setups (e.g., the same number of components and training steps) for Jigsaw, Deep Clustering, and Rotation. Increasing training steps generally improves performance. For Jigsaw and Rotation, starting with Discriminative Encoders yields better results, while for Deep Clustering, starting with reconstructive pretraining is more effective. This phenomenon is attributed to reconstructive pretraining enhancing feature learning for clustering.

Method	Approach	Pretrained components utilized for segmentation	BMS	NCS	Pretrained component utilized for classification	ECC
Jigsaw	(D)	$\mathcal{E}_D$, $D_\emptyset$	63.33 ± 1.11	73.38 ± 1.65	$\mathcal{E}_D$	81.79 ± 1.04
	(R)	$\mathcal{E}_R$, D_R	62.94 ± 0.84	73.17 ± 1.84	$\mathcal{E}_R$	81.05 ± 1.70
	(D+R)	$\mathcal{E}_{(D+R)}$, $D_{(D+R)}$	64.44 ± 0.97	73.6 ± 1.48	$\mathcal{E}_{(D+R)}$	83.84 ± 1.02
	(R+A)	$\mathcal{E}_{(R+A)}$, $D_{(R+A)}$	63.98 ± 0.57	73.46 ± 1.15	$\mathcal{E}_{(R+A)}$	83.07 ± 1.32
	D(D+R)	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	65.33 ± 1.31***	74.53 ± 1.13***	$\mathcal{E}_{D(D+R)}$	84.49 ± 1.38***
	R(R+D)	$\mathcal{E}_{R(R+D)}$, $D_{R(R+D)}$	64.14 ± 0.81	73.71 ± 0.78	$\mathcal{E}_{R(R+D)}$	83.66 ± 1.2
	R(R+A)	$\mathcal{E}_{R(R+A)}$, $D_{R(R+A)}$	64.21 ± 0.97	73.33 ± 0.47	$\mathcal{E}_{R(R+A)}$	83.44 ± 0.93
	(D+R+A)	$\mathcal{E}_{(D+R+A)}$, $D_{(D+R+A)}$	64.98 ± 0.68	74.32 ± 1.32	$\mathcal{E}_{(D+R+A)}$	84.12 ± 1.38
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	65.34 ± 1.13	74.76 ± 1.02	$\mathcal{E}_{D(D+R)(D+R+A)}$	84.73 ± 0.66*
	(R+A)(R+A+D)	$\mathcal{E}_{(R+A)(R+A+D)}$, $D_{(R+A)(R+A+D)}$	64.63 ± 1.69	73.95 ± 1.77	$\mathcal{E}_{(R+A)(R+A+D)}$	83.76 ± 1.4
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	66.07 ± 1.33***	74.87 ± 1.17*	$\mathcal{E}_{D(D+R)(D+R+A)}$	84.89 ± 1.05**
	R(R+D)(R+D+A)	$\mathcal{E}_{R(R+D)(R+D+A)}$, $D_{R(R+D)(R+D+A)}$	65.03 ± 1.24	73.96 ± 0.97	$\mathcal{E}_{R(R+D)(R+D+A)}$	83.94 ± 1.48
	R(R+A)(R+A+D)	$\mathcal{E}_{R(R+A)(R+A+D)}$, $D_{R(R+A)(R+A+D)}$	64.97 ± 0.91	73.57 ± 1.88	$\mathcal{E}_{R(R+A)(R+A+D)}$	84.02 ± 1.7
Deep Clustering	(D)	$\mathcal{E}_D$, $D_\emptyset$	65.81 ± 0.73	74.82 ± 0.47	$\mathcal{E}_D$	84.82 ± 0.62
	(R)	$\mathcal{E}_R$, D_R	63.65 ± 0.41	74.26 ± 0.78	$\mathcal{E}_R$	83.07 ± 0.89
	(D+R)	$\mathcal{E}_{(D+R)}$, $D_{(D+R)}$	64.53 ± 0.81	74.24 ± 1.63	$\mathcal{E}_{(D+R)}$	84.13 ± 1.89
	(R+A)	$\mathcal{E}_{(R+A)}$, $D_{(R+A)}$	64.12 ± 0.93	74.03 ± 1.21	$\mathcal{E}_{(R+A)}$	83.92 ± 1.12
	D(D+R)	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	66.73 ± 0.51	75.91 ± 1.12	$\mathcal{E}_{D(D+R)}$	85.12 ± 1.37
	R(R+D)	$\mathcal{E}_{R(R+D)}$, $D_{R(R+D)}$	67.58 ± 0.34***	76.32 ± 0.78	$\mathcal{E}_{R(R+D)}$	85.41 ± 0.94*
	R(R+A)	$\mathcal{E}_{R(R+A)}$, $D_{R(R+A)}$	67.34 ± 0.91	76.61 ± 0.76***	$\mathcal{E}_{R(R+A)}$	85.28 ± 0.73
	(D+R+A)	$\mathcal{E}_{(D+R+A)}$, $D_{(D+R+A)}$	65.9 ± 1.72	74.66 ± 1.89	$\mathcal{E}_{(D+R+A)}$	84.57 ± 1.66
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	66.8 ± 0.71*	76.18 ± 0.95**	$\mathcal{E}_{D(D+R)(D+R+A)}$	84.59 ± 0.5
	(R+A)(R+A+D)	$\mathcal{E}_{(R+A)(R+A+D)}$, $D_{(R+A)(R+A+D)}$	65.96 ± 1.29	74.51 ± 1.39	$\mathcal{E}_{(R+A)(R+A+D)}$	84.2 ± 0.94
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	67.22 ± 2.33	76.38 ± 1.63	$\mathcal{E}_{D(D+R)(D+R+A)}$	84.82 ± 0.62
	R(R+D)(R+D+A)	$\mathcal{E}_{R(R+D)(R+D+A)}$, $D_{R(R+D)(R+D+A)}$	68.36 ± 1.14**	77.25 ± 1.11**	$\mathcal{E}_{R(R+D)(R+D+A)}$	85.57 ± 1.84
	R(R+A)(R+A+D)	$\mathcal{E}_{R(R+A)(R+A+D)}$, $D_{R(R+A)(R+A+D)}$	68.22 ± 0.86	76.71 ± 1.24	$\mathcal{E}_{R(R+A)(R+A+D)}$	85.69 ± 1.41***
Rotation	(D)	$\mathcal{E}_D$, $D_\emptyset$	65.81 ± 0.73	74.82 ± 0.47	$\mathcal{E}_D$	84.82 ± 0.62
	(R)	$\mathcal{E}_R$, D_R	63.72 ± 0.58	74.35 ± 0.97	$\mathcal{E}_R$	83.23 ± 1.17
	(D+R)	$\mathcal{E}_{(D+R)}$, $D_{(D+R)}$	64.78 ± 0.98	74.37 ± 0.69	$\mathcal{E}_{(D+R)}$	83.12 ± 1.43
	(R+A)	$\mathcal{E}_{(R+A)}$, $D_{(R+A)}$	64.17 ± 0.87	74.11 ± 1.16	$\mathcal{E}_{(R+A)}$	83.83 ± 1.05
	D(D+R)	$\mathcal{E}_{D(D+R)}$, $D_{(D+R)}$	65.44 ± 0.67**	74.86 ± 0.58**	$\mathcal{E}_{D(D+R)}$	83.86 ± 1.12*
	R(R+D)	$\mathcal{E}_{R(R+D)}$, $D_{R(R+D)}$	64.88 ± 0.91	74.61 ± 0.76	$\mathcal{E}_{R(R+D)}$	82.92 ± 0.83
	R(R+A)	$\mathcal{E}_{R(R+A)}$, $D_{R(R+A)}$	64.13 ± 0.43	74.55 ± 0.81	$\mathcal{E}_{R(R+A)}$	83.03 ± 0.59
	(D+R+A)	$\mathcal{E}_{(D+R+A)}$, $D_{(D+R+A)}$	64.92 ± 1.52	74.56 ± 0.97	$\mathcal{E}_{(D+R+A)}$	83.34 ± 1.56
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	65.52 ± 1.19	74.94 ± 0.61	$\mathcal{E}_{D(D+R)(D+R+A)}$	84.12 ± 0.95
	(R+A)(R+A+D)	$\mathcal{E}_{(R+A)(R+A+D)}$, $D_{(R+A)(R+A+D)}$	64.99 ± 0.87	74.58 ± 1.14	$\mathcal{E}_{(R+A)(R+A+D)}$	83.47 ± 1.27
	D(D+R)(D+R+A)	$\mathcal{E}_{D(D+R)(D+R+A)}$, $D_{(D+R)(D+R+A)}$	66.13 ± 0.65*	74.93 ± 0.62*	$\mathcal{E}_{D(D+R)(D+R+A)}$	84.62 ± 1.37**
	R(R+D)(R+D+A)	$\mathcal{E}_{R(R+D)(R+D+A)}$, $D_{R(R+D)(R+D+A)}$	65.24 ± 0.73	74.24 ± 0.91	$\mathcal{E}_{R(R+D)(R+D+A)}$	83.58 ± 0.94
	R(R+A)(R+A+D)	$\mathcal{E}_{R(R+A)(R+A+D)}$, $D_{R(R+A)(R+A+D)}$	65.32 ± 0.88	74.12 ± 0.64	$\mathcal{E}_{R(R+A)(R+A+D)}$	83.62 ± 1.21

* $p < 0.5$.
** $p < 0.1$.
*** $p < 0.05$.

same number of training epochs being performed (Fig. 7). Table 7 indicates that the D(D+R)(D+R+A) strategy performs best for Jigsaw and Rotation in downstream tasks, whereas R(R+D)(R+D+A) and R(R+A)(R+A+D) outperforms the D(D+R)(D+R+A) strategy for Deep

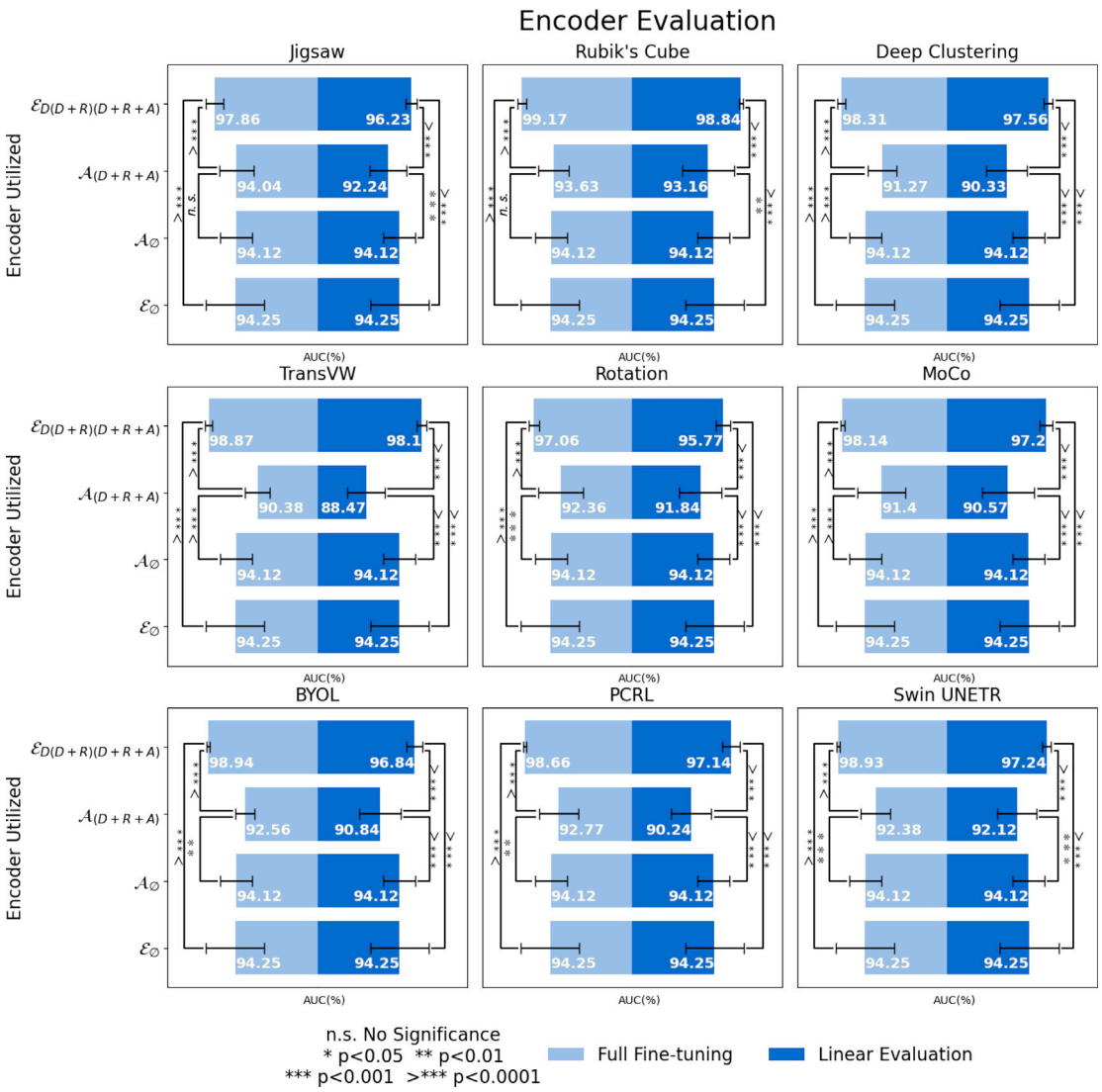


Fig. 8. Adversarial encoders learnt weak representation, and they are not suitable for target tasks. $\mathcal{A}_{(D+R+\mathcal{A})}$ pretrained with all nine SSL methods perform worse than $\mathcal{A}_{\emptyset}$, indicating that their learned representations are not suitable for the target task. By contrast, SSL initialized discriminative encoders ($\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$) all perform better than random initialization ($\mathcal{E}_{\emptyset}$) and $\mathcal{A}_{(D+R+\mathcal{A})}$ for the target task.

Clustering. While incremental pretraining with SDM strategy typically yields the best performance for most methods, Deep Clustering benefits more from the incremental pretraining with SRM strategy. In our extensive experiments, we conclude that for non-clustering types of discriminative methods, it is best practice to use the SDM strategy, while for clustering types of discriminative methods, SRM strategy yields the best performances. We believe this phenomenon is due to the fact that reconstructive pretraining helps the encode learn features more consistent with appearance, thereby enabling clustering.

5.2. Adversarial encoders are not suitable for transfer learning as they learn weak representations

With stepwise incremental pretraining, we obtain two pretrained encoders, $\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$ and $\mathcal{A}_{D+R+\mathcal{A}}$, from the "United" model for target tasks.We evaluate their performance on the task of lung nodule false positive reduction (NCC) with two settings: (1) linear evaluation, which fixes the pre-trained network and uses the features it computes to train a linear classifier for the target task, and (2) full fine-tuning of the pre-trained network for the target task. For linear evaluation, there is a significant performance difference between Encoder $\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$ and Encoder $\mathcal{A}_{D+R+\mathcal{A}}$. As shown in Fig. 8, the adversarial encoders are weaker than discriminative encoders. We believe it is because the only pretraining supervision signal for the adversarial encoders is to distinguish real and fake images. This results in decreased performance for Jigsaw by AUC scores of 4.0%. Similarly, Rubik's Cube decreased by 6.7%, Deep Clustering by 7.2%, TransVW by 9.6%, Rotation by 3.9%, MoCo by 6.6%, BYOL by 6.0%, PCRL by 6.9%, and Swin UNETR by 5.1%. Furthermore, the adversarial encoders' performance is also worse than that of random initialized encoders $\mathcal{A}_{\emptyset}$. This results in decreased performance for Jigsaw by AUC scores of 1.9%, for Rubik's Cube by 1%, for Deep Clustering by 2.8%, for TransVW by 5.7%, for Rotation by 2.3%, for MoCo by 3.6%, for BYOL by 3.3%, PCRL by 3.9%, and Swin UNETR by 2%. It is evident that the fixed features computed by the pretrained Encoder $\mathcal{A}_{D+R+\mathcal{A}}$ do not transfer well for the target task. Even when compared with the randomly initialized Encoder $\mathcal{A}_{\emptyset}$, the computed features become less useful. We further evaluate the two encoders through full fine-tuning. While the Encoder $\mathcal{A}_{D+R+\mathcal{A}}$ improves compared to its evaluation using linear evaluation, it still lags behind Encoder $\mathcal{E}_{D(D+R)(D+R+\mathcal{A})}$. More importantly, the adversarial encoders' performance is not stable compared to discriminative encoders, as their standard deviations are higher.

6. Conclusion

We have developed a United framework that integrates discriminative SSL methods with restorative and adversarial learning. Our extensive experiments demonstrate that our pretrained United models consistently outperform the SoTA baselines. This performance improvement is attributed to our stepwise incremental pretraining scheme, which not only stabilizes the pretraining but also unleashes the synergy of discriminative, restorative, and adversarial learning. We expect that our pretrained United models will exert an important impact on medical image analysis across diseases, organs, modalities, and specialties.

CRediT authorship contribution statement

Zuwei Guo: Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Nahid Ul Islam:** Formal analysis, Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Michael B. Gotway:** Data curation, Funding acquisition, Investigation, Resources, Writing – review & editing. **Jianming Liang:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jianming Liang has patent pending to Arizona State University.

Data availability

The GitHub link is included in the paper.

Acknowledgments

We thank F. Haghighi, M. R. Hosseinzadeh Taher, and Z. Zhou for their discussions, debates, and support in implementing the earlier ideas behind “United & Unified” and in drafting earlier versions. We thank R. Feng for implementing MoCo (He et al., 2020) and conducting the experiments using the stepwise incremental pretraining. This research has been supported in part by ASU, USA and Mayo Clinic, USA through a Seed Grant and an Innovation Grant, and in part by the NIH, USA under Award Number R01HL128785. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. This work has utilized the GPUs provided in part by the ASU Research Computing and in part by the Bridges-2 at Pittsburgh Supercomputing Center through allocation BCS190015 and the Anvil at Purdue University through allocation MED220025 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation, USA grants #2138259, #2138286, #2138307, #2137603, and #2138296. The content of this paper is covered by patents pending.

References

Armato III, S.G., McLennan, G., Bidaut, L., McNitt-Gray, M.F., Meyer, C.R., Reeves, A.P., Zhao, B., Aberle, D.R., Henschke, C.I., Hoffman, E.A., et al., 2011. The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. Med. Phys. 38 (2), 915–931.

Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha, S.M., Rozycki, M., et al., 2018. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. arXiv preprint arXiv:1811.02629.

Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.-W., Han, X., Heng, P.-A., Hesser, J., et al., 2019. The liver tumor segmentation benchmark (lits). arXiv preprint arXiv:1901.04056.

Caron, M., Bojanowski, P., Joulin, A., Douze, M., 2018. Deep clustering for unsupervised learning of visual features. In: European Conference on Computer Vision.

Chen, X., Fan, H., Girshick, R., He, K., 2020. Improved baselines with momentum contrastive learning. arXiv:2003.04297.

Chen, X., He, K., 2021. Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 15750–15758.

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. arXiv:1803.07728.

Grill, J.-B., Strub, F., Altche, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M., 2020. Bootstrap your own latent: A new approach to self-supervised learning. arXiv:2006.07733.

Haghighi, F., Taher, M.R.H., Gotway, M.B., Liang, J., 2022. DiRA: Discriminative, restorative, and adversarial learning for self-supervised medical image analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20824–20834.

Haghighi, F., Taher, M.R.H., Gotway, M.B., Liang, J., 2024. Self-supervised learning for medical image analysis: Discriminative, restorative, or adversarial? Med. Image Anal. 103086.

Haghighi, F., Taher, M.R.H., Zhou, Z., Gotway, M.B., Liang, J., 2021. Transferable visual words: Exploiting the semantics of anatomical patterns for self-supervised learning. IEEE Trans. Med. Imaging 1. http://dx.doi.org/10.1109/TMI.2021.3060634.

Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H.R., Xu, D., 2021. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: International MICCAI Brainlesion Workshop. Springer, pp. 272–284.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738.

Jing, L., Tian, Y., 2020. Self-supervised visual feature learning with deep neural networks: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (11), 4037–4058.

Liang, J., Bi, J., 2007. Computer aided detection of pulmonary embolism with tobogganing and mutiple instance classification in CT pulmonary angiography. Inf. Process. Med. Imaging 20, 630–641.

Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al., 2014. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging 34 (10), 1993–2024.

Noroozi, M., Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. CoRR arXiv:1603.09246, URL http://arxiv.org/abs/1603.09246.

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. Springer, pp. 234–241.

Setio, A.A.A., Traverso, A., De Bel, T., Berens, M.S., Van Den Bogaard, C., Cerello, P., Chen, H., Dou, Q., Fantacci, M.E., Geurts, B., et al., 2017. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. Med. Image Anal. 42, 1–13.

Siddique, N., Sidike, P., Elkin, C., Devabhaktuni, V., 2020. U-Net and its variants for medical image segmentation: theory and applications. URL http://arxiv.org/abs/2011.01118.

Tajbakhsh, N., Gotway, M.B., Liang, J., 2015. Computer-aided pulmonary embolism detection using a novel vessel-aligned multi-planar image representation and convolutional neural networks. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, Cham, pp. 62–69.

Tajbakhsh, N., Roth, H., Terzopoulos, D., Liang, J., 2021. Guest editorial annotation-efficient deep learning: The holy grail of medical imaging. IEEE Trans. Med. Imaging 40 (10), 2526–2533.

Tajbakhsh, N., Shin, J.Y., Gotway, M.B., Liang, J., 2019. Computer-aided detection and visualization of pulmonary embolism using a novel, compact, and discriminative image representation. Med. Image Anal. 58, 101541.

Taleb, A., Loetzsch, W., Danz, N., Severin, J., Gaertner, T., Bergner, B., Lippert, C., 2020. 3D self-supervised methods for medical imaging. Adv. Neural Inf. Process. Syst. 33, 18158–18172.

Tang, Y., Yang, D., Li, W., Roth, H.R., Landman, B., Xu, D., Nath, V., Hatamizadeh, A., 2022. Self-supervised pre-training of swin transformers for 3d medical image analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20730–20740.

Tao, X., Li, Y., Zhou, W., Ma, K., Zheng, Y., 2020. Revisiting rubik’s cube: Self-supervised learning with volume-wise transformation for 3D medical image segmentation. arXiv:2007.08826.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S., 2021. Barlow twins: Self-supervised learning via redundancy reduction. arXiv:2103.03230.

Zhou, H.-Y., Lu, C., Yang, S., Han, X., Yu, Y., 2021a. Preservational learning improves self-supervised medical image models by reconstructing diverse contexts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3499–3509.

Zhou, Z., Sodha, V., Pang, J., Gotway, M.B., Liang, J., 2021b. Models genesis. Med. Image Anal. 67, 101840. http://dx.doi.org/10.1016/j.media.2020.101840.

Zhuang, X., Li, Y., Hu, Y., Ma, K., Yang, Y., Zheng, Y., 2019. Self-supervised feature learning for 3D medical images by playing a rubik’s cube. arXiv:1910.02241.