Pattern Recognition Letters 186 (2024) 64–70

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

DeepSet SimCLR: Self-supervised deep sets for improved pathology representation learning

David Torpey *, Richard Klein

School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa

ARTICLE INFO

Editor: Jiwen Lu

Keywords:
Deep learning
Self-supervised learning
Representation learning
Medical image analysis

ABSTRACT

Often, applications of self-supervised learning to 3D medical data opt to use 3D variants of successful 2D network architectures. Although promising approaches, they are significantly more computationally demanding to train, and thus reduce the widespread applicability of these methods away from those with modest computational resources. Thus, in this paper, we aim to improve standard 2D SSL algorithms by modelling the inherent 3D nature of these datasets implicitly. We propose two variants that build upon a strong baseline model and show that both of these variants often outperform the baseline in a variety of downstream tasks. Importantly, in contrast to previous works in both 2D and 3D approaches for 3D medical data, both of our proposals introduce negligible additional overhead in terms of parameter complexity. Although data loading overhead increases over the baseline SimCLR model (which we can show can be somewhat mitigated through parallelisation), our proposed models are still significantly more efficient than previous approaches based on sequence modelling. Overall, our proposed methods help improve the democratisation of these approaches for medical applications.

1. Introduction

Self-supervised learning (SSL) has pushed forward the capabilities of large-scale visual representation learning due to its ability to model large datasets without manual or human annotation [1–4]. Instead, these methods derive a supervision signal from the data itself and often are architected to solve a proxy task of some type. Through recent improvements, self-supervised pretraining has been shown to outperform supervised pretraining on many downstream tasks [2,5]. This is particularly important for medical domains, where annotation is extremely expensive, laborious, and time-consuming to obtain. However, these methods, even in their standard 2D setting, are often notably computationally intensive to train, often requiring tens of GPUs to train [3,6–8] and sometimes requiring hundreds [2]. 3D extensions of these techniques are even more computationally intensive. It should be noted that even tens of GPUs severely reduce the potential widespread applicability of these techniques.

Motivated by the success of the DeepSet [9] architecture for the application of whole slide image (WSI) classification [10], we propose two variants that build upon a strong modern contrastive SSL baseline. The variants use an intra-scan sampling strategy, with one variant employing the DeepSet architecture, to implicitly model the 3D nature of medical data using an inherently 2D architecture. Whereas previous approaches focus on modelling the 3D volumes using either 3D native architectures [11,12] or computationally-intensive 2D architectures [13], we instead focus on making this process more computationally efficient using an architecture proven to work well for the modelling of pathology data [10] while inducing limited parameter overhead (although our method induces additional data preprocessing complexity, which can be partially offset by parallelised data loading).

We propose two novel variants of a multi-view contrastive baseline: SimCLR [1]. First, we devise a simple and effective slice sampling strategy for the generation of random views. Next, we propose a new architecture where each ‘view’ is a set of images rather than a single image, which is operated upon using a Deep Set network. Our adaptations introduce negligible additional computational cost to the architecture (at most a small multi-layer perceptron). In this way, we inherit the plethora of benefits from the base SimCLR architecture, while improving performance in the medical domain by implicitly modelling 3D medical data. Our proposal models enable applications to 2D downstream tasks (which is not possible for most native 3D architecture).

Through extensive experiments on various downstream datasets and task types, we show that the two variants consistently outperform the baseline, sometimes by a large margin. Further, we perform analyses

* Corresponding author.
E-mail addresses: 674425@students.wits.ac.za (D. Torpey), richard.klein@wits.ac.za (R. Klein).

https://doi.org/10.1016/j.patrec.2024.09.005
Received 24 February 2024; Received in revised form 4 September 2024; Accepted 5 September 2024
Available online 12 September 2024

of core architectural decisions, including the size of the set fed into the Deep Set, the sampling strategy, and how the Deep Set is embedded within SimCLR.

We note that our goal is to improve upon a standard SSL baseline model, and *not to achieve a new state-of-the-art*. This is because the ubiquity of SSL methods means that any improvements can greatly benefit many domains, particularly those tailored for the medical domain due to its highly prohibitive annotation bottleneck. Our focus is on changes to an SSL baseline that are easy to integrate and deploy, while being robust and performant with negligible performance overhead. This is in stark contrast to existing works for self-supervision on medical data which are typically computationally intensive compared to standard SSL models [11–13]. As such, we note our contributions: (1) Two proposed models that implicitly model the 3D nature of pathology data while remaining computationally efficient (parameter complexity) as, and significantly more performant than the strong baseline. (2) A detailed study of the effect of the key introduced hyperparameters. (3) A comprehensive evaluation on a large set of downstream datasets covering a variety of medical tasks and pathology modalities.

The rest of this paper is structured as follows. In Section 2, we describe the relevant literature in both SSL and its applications to the medical domain. Section 3, we describe the baseline model and our two proposed variants in detail. Section 4 contains the experiments and analyses for our variants on multiple downstream datasets and tasks. Lastly, Section 5 contains concluding remarks and proposals for future work.

2. Related work

2.1. Self-supervised learning

Modern approaches to SSL typically fall into one of 3 categories: (1) contrastive, (2) non-contrastive, or (3) clustering. Techniques such as SimCLR [1,2], NNCLR [14], and MoCo [4,15] are contrastive approaches. These approaches aim to pull representations of similar 'views' (i.e. those sourced from the same image) together while pulling the representations of dissimilar 'views' apart. Non-contrastive approaches such as Barlow Twins [6] and BYOL [7] still discriminate between instances but instead optimise for different objectives that are not reliant on negative samples such as contrastive losses. For example, Barlow Twins aims to make the cross-correlation matrix between two batches of representations of views to be as similar to the identity matrix as possible. BYOL is a self-distillation approach that aims to train a teacher–student architecture where the goal of the first network is to predict the representation produced by the other.

Clustering approaches such as SwAV [8] try to circumvent a strong assumption made by contrastive approaches where views from the same image, whatever they may depict, are trained to be represented by the same vector, and views from different images are always trained to be as dissimilar as possible. SwAV therefore attempts to discriminate between clusters of representations instead of individual instance representations, where clusters contain representations of different views of the same image.

In general, these types of SSL approaches rely on generating the views through data augmentation using a stochastic augmentation function which is typically comprised of transformations such as random cropping, horizontal flipping, Gaussian blurring, and colour jittering. This augmentation function must produce a variety of views while maintaining the semantics between views generated from the same image. There is also a trend towards attempting to better understand the properties of these SSL methods, as they as not yet well understood [16–18].

Another class of SSL techniques are pretext tasks where the network solves manual proxy tasks usually based on either image restoration or spatial content. These include techniques such as rotation prediction [19], colourisation [20], inpainting [21], and image jigsaw puzzles [22].

2.2. Applications of SSL to medical data

Since SSL is a promising and scalable approach to modelling large datasets while being annotation-free, attempts have been made to apply it to the medical domain where obtaining annotations is particularly prohibitive to do at scale. Most SSL approaches in the medical domain can be classed as either 2D or 3D — whether they model 2D medical image data, or 3D volumes from sources such as computed tomography (CT) or medical resonance imaging (MRI) scans.

Kang et al. [23] show that domain-aligned pretraining (i.e. pretraining on a medical dataset) consistently outperforms the standard paradigm of ImageNet pretraining for multiple common SSL methods. Further, they propose an altered set of transformations for the data augmentation pipeline to generate views more aligned with the medical domain. These changes include weaker colour jittering, random vertical flips, and a novel stain augmentation (tailored to WSI images). This new set of transformations improves performance over the standard set of augmentations included in most modern SSL techniques [1,6,7].

Taleb et al. [11] proposes 3D variants of multiple successful 2D SSL algorithms, including CPC [24], RotNet [19], image jigsaw puzzles [22], relative patch prediction [25], and Exemplar networks [26]. These SSL models are naturally extended into 3D due to their reliance on spatial context. They demonstrate their success for multiple downstream medical tasks.

Tang et al. [12] proposed a novel 3D vision transformer (ViT) [27] architecture to facilitate SSL on 3D medical volumes. They propose random cutout and rotation as the transformations for the generation of 3D views as they posit that these are more aligned with human anatomy. This SS ViT is tasked with solving a multi-task objective consisting of inpainting, rotation prediction, and contrastive learning.

Nguyen et al. [13] attempts to jointly model 2D and 3D medical data to capture both the spatial semantics of the individual 2D images, as well as long-range dependencies between the slices from a 3D volume. The framework is built upon clustering-based SSL algorithms SwAV [8] and Deep-ClusterV2 [28], and is known as Intra-Inter Contrastive Clustering (*IICC*). A transformer [29] with a novel deformable self-attention mechanism [30] is added into the base architecture to capture the aforementioned long-range dependencies of 3D volume data. Our proposed method differs from this approach in that we model the 3D volume as a set or implicitly through sampling, instead of as a sequence with a transformer. Further, our proposed models introduce significantly less computational overhead (both in terms of parameter complexity and training time) than the additional transformer and its masked embedding prediction task. We emphasise that, although modelling a 3D volume as a sequence is perhaps more appropriate, we show that it comes at the penalty of significant computational overhead over modelling the volume as a set, which being outperformed in many downstream tasks. We note that our proposal induces additional data loading overhead over baselines (although as shown in the experiments, these effects are mostly mitigated through parallelisation). However, our proposed models are still significantly more performant (running time) than previous approaches, in particular, IICC.

3. Methodology

Consider a dataset of 3D volumes, such as CT or MRI scans. Each scan S_i is comprised of an *ordered* collection of slices: $S_i = \{x_j \in \mathbb{R}^{H\times W\times C}\}_{j=1}^{L_i}$, where $H \times W \times C$ are the spatial dimensions of slice x_j and L_i is the number of slices in scan S_i. We define a dataset of N scans $\mathcal{S} := \{S_i\}_{i=1}^{N}$ and a flattened version consisting of all slices from all scans $\mathcal{S}_{\texttt{flat}} := \cup S_i$. Note that $|\mathcal{S}_{\texttt{flat}}| = \sum_{i=1}^{N} L_i$.

Define an encoder network f and projection head g, which are the equivalent functions as defined in SimCLR [1]. Finally, we define a (possibly infinite) set of random augmentation functions $\mathcal{A}$, where each $\alpha \in \mathcal{A}$ is a function that applies a pipeline of random transformations to generate views. We use the same base set of transformations as Chen et al. [1] in order to construct $\mathcal{A}$.

3.1. SimCLR baseline

Our baseline model is the standard SimCLR approach (see Figure A.4 for an architecture diagram). A slice is randomly sampled across all the slices from all scans: $x \in S_{\texttt{flat}}$. We generate views using two random augmentation functions: $x_1 = \alpha_1(x)$ and $x_2 = \alpha_2(x)$ where $\alpha_1, \alpha_2 \in \mathcal{A}$. These views are then sent through the encoder and projection head to compute latent representations: $z_1 = g(f(x_1))$ and $z_2 = g(f(x_2))$. The contrastive NT-Xent loss is then computed using these representations. In this way, a positive pair of views is always sourced from the same input slice. This precludes the ability to model the 3D nature of the scans as the model never compares regions from different slices within the same scan.

Instead, it attempts to pull the representations of different views of the same slice together in latent space and push the representations of views from different slices (*even if from different slices potentially within the same scan*) apart. It is unclear apriori whether this assumption is detrimental to transfer learning performance, however, it is the standard SSL contrastive setup, and we make novel adaptations to it to build our two proposed variants: *Per-scan SimCLR* and *DeepSet SimCLR*.

3.2. Per-scan SimCLR

The intuition for our Per-scan SimCLR (PS-SimCLR) approach is that the sampling strategy of the baseline precludes the model from accounting for the 3D nature of the scans. We address this by proposing a simple and effective novel sampling strategy that attempts to implicitly encode the 3D nature of the scans during the modelling process (see Figure A.5 for an architecture diagram). In this way, PS-SimCLR changes how random views are generated but keeps the network architecture consistent with the baseline.

Our sampling strategy is controlled by a sampling width $\omega \in (0, 1]$. This parameter controls the width of a contiguous region (i.e. window) of the scan $s_i \in X$ from which slices can be sampled, as well as the amount of shared semantic and mutual information between random views. We randomly choose the start location j of this window from the range $\{0, \dots, (1-\omega)L_i\}$. The end location of the window is then given by $j+\omega L_i$. The positive pair of random views is then generated by sampling two slices from this window of the scan: $x, x' \in \{x_j, \dots, x_{j+\omega L_i}\}$. The positive view is pair is then defined as $x_1 = \alpha_1(x)$ and $x_2 = \alpha_2(x')$. The rest of the PS-SimCLR pipeline is identical to the baseline: we compute the NT-Xent loss from representations $z_1 = g(f(x_1))$ and $z_2 = g(f(x_2))$. Note that we default to the baseline strategy of generating views if L_i is sufficiently small, i.e. $L_i < T$ for some threshold T (this happens infrequently).

In contrast to the baseline model, PS-SimCLR pulls representations of different views of *any* slice within the same scan together and contrasts them against representations of slices from any other scan. In this way, we implicitly model the 3D nature of the volume by imposing that views of the same volume should be close in latent space, instead of only views of a single slice of a scan needing to be close in latent space.

3.3. DeepSet SimCLR

Our DeepSet SimCLR (DS-SimCLR) variant adds a more explicit mechanism to model the 3D nature of the scans in addition to the sampling strategy used in PS-SimCLR. We model a random view as a set of slices instead of as an individual slice (as in PS-SimCLR and the baseline). Motivated by the success of the Deep Set architecture for supervised WSI classification [10], we propose a novel combination of the Deep Set architecture and SimCLR (see Fig. 1).

Similarly to PS-SimCLR, we use ω to construct the window of a scan S_i with start and end locations j and $j+\omega L_i$. We sample K equidistant images from this window: $\{x_l^{(i)}\}_{l=1}^{K}$. The augmentation functions within $\mathcal{A}$ are constructed using the same base transformations as in PS-SimCLR and the baseline, however, *the random crop occurs at the same location for all K sampled slices in the set*. This is done to ensure semantic consistency of the slices in the set. We then compute a random view X_1 of the scan: $X_1 := \{\alpha_1(x_l^{(i)})\}_{l=1}^{K}$ for $\alpha_1 \in \mathcal{A}$. We construct the other random view X_2 using a different random window of the scan. Note that for DS-SimCLR, random views are defined as a **set** of transformed slices, unlike PS-SimCLR or the baseline where views are individual transformed slices.

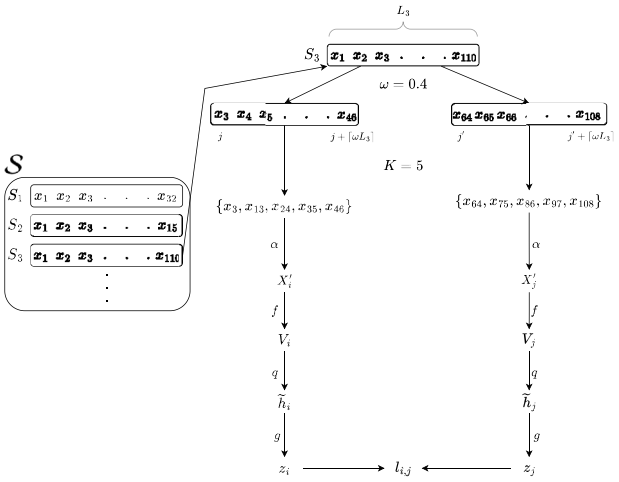


Fig. 1. DS-SimCLR architecture.

To compute latent representations for these set-based random views, we employ the Deep Set architecture. We project each transformed slice of X_1 and X_2 using f: $V_1 = \{f(x)|x \in X_1\}_{l=1}^{K}$ (similarly for X_2: $V_2 = \{f(x)|x \in X_2\}_{l=1}^{K}$). Next, we sum the representations of the set's slices and send them through an MLP, formally (as per Zaheer et al. [9]):

$$\tilde{h}_1 = g_{\texttt{DS}}\left(\sum_{j=1}^{K} V_{1,j}\right) \tag{1}$$

where $g_{\texttt{DS}}$ is a function from the Deep Set architecture used to compute the representation of a set $\tilde{h}_1$. We do the same for V_2 to compute $\tilde{h}_2$. Finally, we compute the final latent representations from which we compute the NT-Xent loss: $z_1 = g(\tilde{h}_1)$ and $z_2 = g(\tilde{h}_2)$. Note that we experiment with different parameterisations of $g_{\texttt{DS}}$, including the identity function and an MLP.

The use of the DeepSet architecture means that slices of a scan are modelled in a permutation invariant way since we are not modelling the scan's 3D volume as a sequence, but instead as a set. This means we trade off the ability to model a scan as a sequence in favour of significantly lower computational demand. Further, Deep Sets have previously been shown as a promising way of modelling medical data [10]. Note that we still do indeed incorporate the inherent 3D nature of the volume into our modelling process by treating it as a set. We ensure that subvolumes (along the axial plane) of the same scan are pulled together, and subvolumes from different scans are pushed apart. Note that if $K = 1$ and $g_{\texttt{DS}} = \texttt{id}$, our DS-SimCLR model is equivalent to the baseline model.

3.4. Implementation details

3.4.1. Datasets

We pretrain on the NLST dataset, which consists of 191133 CT scans of human lungs. We use a random subset of 2000 scans to align with the size of pretraining datasets used in previous works [13] and for computational reasons. For downstream evaluation, we use the PCAM, CRC, CovidX-CT, and CheXpert (henceforth denoted as CX) datasets for

Table 1
Linear evaluation results of the baseline and two proposed variants, PS-SimCLR and DS-SimCLR, for various classification datasets.

	CRC	CovidX-CT	PCAM	CX (Atelectasis)	CX (Cardiomegaly)	CX (Edema)
Standard	82.27 ± 0.17	79.68 ± 0.40	76.55 ± 0.34	35.38 ± 0.58	60.57 ± 0.66	58.69 ± 0.51
PS-SimCLR	83.69 ± 0.44	82.02 ± 0.37	79.22 ± 0.30	34.79 ± 0.29	**62.34 ± 0.84**	**61.57 ± 1.06**
DS-SimCLR	**89.13 ± 0.21**	**84.84 ± 0.45**	**81.14 ± 0.56**	**35.98 ± 1.15**	56.67 ± 0.54	58.97 ± 0.93
IICC	69.89 ± 0.90	75.78 ± 0.71	78.22 ± 0.34	34.19 ± 0.00	57.92 ± 1.02	53.02 ± 1.06

the linear evaluation task. We use the Kvasir Instruments and TBX11 datasets for downstream fine-tuning for the semantic segmentation (SemSeg) and object detection (OD) tasks, respectively. These datasets provide a variety of pathology tasks to ensure a comprehensive evaluation of our pretrained models. The performance metrics are accuracy for linear evaluation, mean average precision for object detection, and mean intersection-over-union for semantic segmentation. We run 5 or 15 random trials for each dataset (depending on the dataset) and report 95% confidence intervals.

3.4.2. Architecture

We parameterise f as a ResNet50 [31] and g as the standard two-layer non-linear MLP in all cases. $g_{\mathtt{DS}}$ is parameterised as a one-layer non-linear MLP. Note that we experiment with an alternate parameterisation of $g_{\mathtt{DS}}$ as the identity function, denoted `id`.

3.4.3. Optimisation

In all cases, we train with an initial learning rate of 0.07 (following the learning rate formula from [1]) and implement cosine learning rate decay throughout training. We set the weight decay to $1e-10$, batch size to 32, and train for 100 epochs using the Adam [32] optimiser.

3.4.4. Key hyperparameters

We train with an image size of 224 × 224 for the baseline and PS-SimCLR, and 128 × 128 for DS-SimCLR by default (to reduce computational complexity). For PS-SimCLR and DS-SimCLR, we set $\omega = 0.1$ and $\omega = 0.5$, respectively (unless otherwise specified). We set $T = 5$ in PS-SimCLR, and $K = 3$ by default in DS-SimCLR. Note that we experiment with other values for these hyperparameters in our analyses.

3.4.5. Prior art

Since we focus on 2D downstream medical tasks, comparing our approach with methods that work on such tasks is most appropriate (importantly, 3D-native architectures preclude evaluation on 2D downstream tasks [13], and therefore are not considered here). We, thus, compare to IICC (Section 2), since this is the latest state-of-the-art method for pretraining on 3D medical data while enabling evaluation on 2D medical data. We compare our method to IICC in terms of both pretraining efficiency and downstream performance.

4. Experiments

4.1. Linear evaluation on downstream datasets

From Table 1, we can see that our two proposed variants – PS-SimCLR and DS-SimCLR – consistently outperform the baseline for all linear evaluation datasets, with DS-SimCLR outperforming both by a large margin on the CRC, CovidX-CT, and PCAM datasets (statistically significant with $p < 0.05$). For example, DS-SimCLR shows an 8.3% and 6.1% increase in performance for the CRC and CovidX-CT datasets compared to the baseline. Overall, DS-SimCLR shows a notable average relative increase in performance of 3.1% across these datasets, and 3.0% for PS-SimCLR. This suggests that implicitly encoding the 3D nature of volumes during pretraining assists with downstream linear evaluation performance, even for such tasks, which are not inherently 3D. This is also particularly notable because DS-SimCLR is trained on lower-resolution random views than the baseline and PS-SimCLR in these experiments.

Table 2
Fine-tuning results of the baseline and two proposed variants, PS-SimCLR and DS-SimCLR, for the TBX11 OD task and Kvasir Instruments SemSeg task.

	TBX11	Kvasir Instruments
Standard	23.09 ± 0.41	78.99 ± 0.43
PS-SimCLR	23.03 ± 0.41	80.12 ± 0.74
DS-SimCLR	23.01 ± 0.49	81.03 ± 0.75
IICC	**25.19 ± 0.87**	**90.10 ± 0.12**

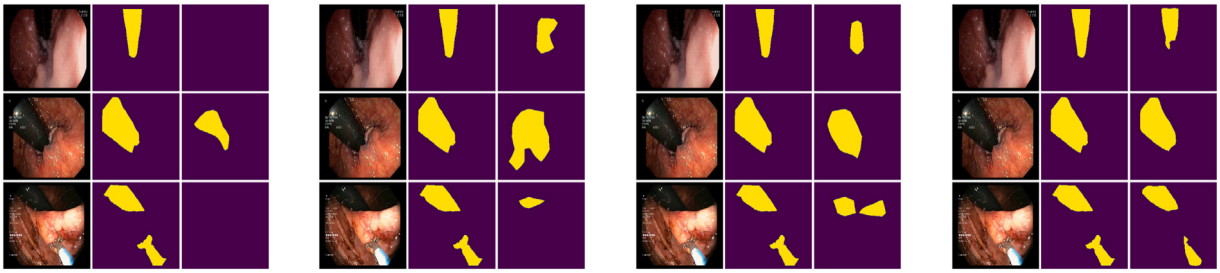
Our proposed methods are also fairly robust to the modality of the pathology in the downstream task. Pretraining was performed using the NLST datasets, which consists of 3D CT scans, and both variants perform well on downstream linear evaluation tasks consisting of X-rays (CheXpert), RGB histopathologic scans (CRC, PCAM), and CT scans (CovidX-CT). Interestingly, the results suggest that PS-SimCLR performs better than DS-SimCLR on 2 of the 3 studied CheXpert X-ray tasks. We posit that the stronger bias imposed upon DS-SimCLR regarding modelling the 3D nature of the pretraining data compared to PS-SimCLR's means that performance is better on this particular inherently 2D modality. It should be noted, however, that this does not reflect similarly in the inherently 2D tasks of CRC and PCAM. Finally, we note our improved performance within the linear evaluation paradigm suggests that the representations learned from our proposed methods better encode the domain-related information to make them more effective downstream as compared to the baseline. We posit this is mainly due to the implicit 3D modelling of PS-SimCLR's sampling strategy and DS-SimCLR's DeepSet component.

Further, we find our methods outperform IICC significantly on most of these linear evaluation tasks. This is notable since IICC is significantly more computationally intensive to train (Section 4.5). This suggests that modelling a volume as a sequence is no more beneficial than modelling it as a set **for classification tasks**, and sequence modelling induces a nontrivial additional computational burden.

4.2. Fine-tuning on downstream datasets

Table 2 shows the results of fine-tuning for two important medical tasks: OD and SemSeg. Note that in this setting, the encoder network f changes its weights during training. Similar to the linear evaluation results shown in Table 1, both of our PS-SimCLR and DS-SimCLR proposed variants outperform the standard SSL baseline. Importantly, IICC outperforms the proposed approaches on both datasets. We attribute this to the difference in how IICC and DS-SimCLR model the 3D volumes. As previously discussed, IICC more appropriately models the volume as a sequence, which we posit enables more efficient and accurate encoding of dense information that better supports dense tasks, such as segmentation or detection. Modelling as a set likely does not do this as efficiently. Notably, however, we see that DS-SimCLR is significantly more computationally tractable and efficient to train than IICC, and this trade-off should be considered on a per-application basis.

Fig. 2 visualises the SemSeg results for the 3 models. The predicted masks for DS-SimCLR are notably more coherent and accurate than both the baseline and PS-SimCLR. The standard model completely fails to detect the instrument in two of these random images. Further, PS-SimCLR is consistently able to detect that an instrument is present in



(a) Standard. (b) PS-SimCLR. (c) DS-SimCLR. (d) IICC.

Fig. 2. Visualisation of SemSeg fine-tuning results on the Kvasir Instruments dataset for 3 randomly chosen test images. The three columns are L-R: input image, ground-truth mask, and predicted mask.

Table 3
Results of DS-SimCLR for various set sizes K. We set $\omega = 0.5$, $D = 64$, and $g_{\text{DS}} = \texttt{id}$ for these experiments. We sample three equidistant values for K.

	CRC	CovidX-CT	PCAM	CX (Atelectasis)	CX (Cardiomegaly)	CX (Edema)	TBX11	Kvasir
$K = 1$	77.53	77.34	79.06	34.27	54.87	53.93	20.53	75.28
$K = 3$	88.15	**84.6**	**80.83**	34.7	**60.09**	57.61	**22.19**	**78.56**
$K = 5$	**89.26**	83.78	78.83	**36.32**	58.72	**59.32**	20.39	75.52

the image, however, the localisations are not as accurate as DS-SimCLR. These results suggest that both proposed variants, and DS-SimCLR in particular, outperform the baseline in both sparse prediction tasks (linear evaluation) and these dense prediction tasks. Notably, IICC is able to much more accurately localise the instruments than the other methods, suggesting that modelling the slices as a sequence helps more for dense tasks than modelling as a set, and less so for the classification tasks in Section 4.1.

4.3. Impact of set size and sampling width

Next, we perform experiments to analyse how the most important hyperparameters introduced by two proposed methods affect performance. For PS-SimCLR, we focus on the sampling width ω. We note that for PS-SimCLR, ω has a bigger impact on performance than T, and thus focus on it here. See Appendix B.6 for more experiments around hyperparameter T. For DS-SimCLR, we also analyse on sampling width ω. Further, we include analyses of the set size K, as this is a core hyperparameter for the DeepSet architecture within our proposed framework.

4.3.1. Set size

First, we analyse the impact of the set size K on the performance of our DS-SimCLR model. We analyse three equally-spaced values for K while keeping other hyperparameters constant. We use the values $K \in \{1, 3, 5\}$ to keep the computational complexity comparable with the baseline. Results can be seen in Table 3. When $K = 5$, linear evaluation (i.e. the classification datasets) performance is highest overall. However, the fine-tuning, dense prediction tasks of TBX11 and Kvasir Instruments result in the best performance for $K = 3$. Interestingly, $K = 1$ is consistently the worst-performing setting.

We hypothesise that the improved performance of $K = 3$ over $K = 5$ for fine-tuning, dense prediction tasks is because dense features may be harder to encode during learning when K is larger since the variance in the set grows accordingly. When the variance is larger, two very distinct sets of images are encouraged to be similar through the contrastive loss. Thus, detailed semantic information is lost through the aggregation of the distinct set of images. Similarly, when K is too small (i.e. $K = 1$), we completely lose the benefit of the DeepSet and associated image set comparison, and therefore performance degrades.

4.3.2. Sampling width

The sampling width ω controls the width of the contiguous window of the scan from which we can sample slices for PS-SimCLR and DS-SimCLR. Similar to the set size analysis, we vary ω while keeping all other hyperparameters fixed. We test a small, medium, and large value for ω in both cases. Results for both variants can be seen in Table 4. We see that for PS-SimCLR, $\omega = 0.1$ results in the best performance overall, results in the best performance in 6/8 of the downstream tasks. In these two other tasks, the performance difference is small. We posit that this is because as $\omega \to 1$, the mutual information between the two sampled slices decreases. As the mutual information decreases, the semantic consistency between the two images decreases, which makes the contrastive task harder.

For DS-SimCLR, we note that overall, $\omega = 0.5$ performs best (for 4/8 tasks), however, there are 2 tasks each where $\omega = 0.8$ and $\omega = 0.2$ perform best. The reason there is no clear winner is because ω has less of an effect on the representations learned by the DeepSet architecture compared with the set size K due to the equidistant sampling utilised to create the set. One can still achieve a balance of image variety and mutual information by sampling more images within the same region. It is clear from these results that $\omega = 0.5$ provides a good balance of mutual information within the set such that useful representations can be learned during pretraining for a variety of downstream tasks.

4.4. Analysis of g_{DS}

Next, we analyse the effect of different parameterisations of g_{DS} in Eq. (1). If we were to make g_{DS} the identity function (i.e. $g_{\text{DS}}(x) = x$ $\forall x$), then we would leave it to the projection head g to serve as the DeepSet's proxy MLP. This slightly reduces computational overhead by negating the need for a forward pass through a small MLP. Results for both the identity and MLP parameterisations of g_{DS} can be seen in Table 5.

The results suggest that for certain parameterisations of DS-SimCLR, there is a clear performance increase, on average, when an MLP is used (such as in $\texttt{DS} - \texttt{SimCLR}(3, 128, 0.5)$ and $\texttt{DS} - \texttt{SimCLR}(2, 224, 0.7)$). However, for different parameterisations, the identity function results in superior performance, such as in $\texttt{DS} - \texttt{SimCLR}(5, 64, 0.5)$ and $\texttt{DS} - \texttt{SimCLR}(5, 64, 0.8)$. These results suggest that the choice of g_{DS} is heavily influenced by the choice of the remaining parameters K D, and ω. Interestingly, for the SemSeg downstream task (Kvasir), the best performance is always achieved when an MLP is used.

Moreover, if we aggregate the results by dataset based on g_{DS}'s parameterisation (see Table C.7 in the Appendix), we see that for the majority of downstream tasks, an MLP performs best on average. This suggests that decoupling the DeepSet's MLP from the projection head, and thus from the SimCLR architecture, results in better performance. However, we reiterate that this is largely dependent on the hyperparameter choices.

Table 4
Results of PS-SimCLR and DS-SimCLR for various sampling widths. For PS-SimCLR, we set $T = 5$. For DS-SimCLR, we set $K = 3$, $D = 64$, and $g_{\text{DS}} = \texttt{id}$. We sample three equidistant values for ω for each model.

		CRC	CovidX-CT	PCAM	CX (Atelectasis)	CX (Cardiomegaly)	CX (Edema)	TBX11	Kvasir
PS-SimCLR	$\omega = 0.1$	**83.69**	**81.96**	**79.22**	34.79	62.91	**61.71**	**23.1**	**80.12**
	$\omega = 0.4$	79.72	78.26	76.79	34.96	61.03	54.7	22.32	79.64
	$\omega = 0.7$	79.55	78.43	77.63	**35.3**	**63.59**	58.72	22.82	78.24
DS-SimCLR	$\omega = 0.8$	88.79	84.59	**79.93**	35.38	56.15	57.35	**21.9**	76.23
	$\omega = 0.5$	**89.26**	83.78	78.83	**36.32**	**58.72**	**59.32**	20.39	75.52
	$\omega = 0.2$	88.75	**85.13**	79.61	34.62	56.15	58.03	21.12	**78.32**

Table 5
Results for the inclusion or exclusion of an MLP in the DeepSet formulation (see Eq. (1)). Note that in $\texttt{DS-SimCLR}(K, D, \omega)$, K is the set size, D is the image size (height and width), and ω is the sampling width.

	g_{DS}	CRC	CovidX-CT	PCAM	CX (Atelectasis)	CX (Cardiomegaly)	CX (Edema)	TBX11	Kvasir
DS(5, 64, 0.5)	id	**89.26**	83.78	**78.83**	**36.32**	**58.72**	**59.32**	**20.39**	75.52
DS(5, 64, 0.5)	MLP	88.82	**84.5**	78.79	34.79	52.48	57.09	20.32	**79.07**
DS(5, 64, 0.8)	id	88.79	84.59	79.93	**35.38**	56.15	**57.35**	**21.9**	76.23
DS(5, 64, 0.8)	MLP	**88.92**	**85.56**	**80.55**	34.7	**60.68**	55.81	21.09	**77.28**
DS(3, 128, 0.5)	id	88.53	84.07	**81.23**	35.38	**58.55**	57.09	22.47	79.27
DS(3, 128, 0.5)	MLP	**89.13**	**84.72**	81.14	**35.98**	56.75	**59.06**	**23.37**	**81.03**
DS(2, 224, 0.7)	id	**83.08**	**82.91**	78.58	**35.13**	62.05	56.5	20.4	75.73
DS(2, 224, 0.7)	MLP	82.82	81.61	**79.35**	35.04	**64.62**	**60.0**	**22.65**	**78.17**

4.5. Complexity analysis

To better understand and quantify the computational efficiency of our proposed DS-SimCLR model versus prior works, we analyse and compare it against the IICC model through the lens of a formal complexity analysis and empirical computation time profiling. For more detailed results, see Appendix E and Appendix F. Recall that one of the core differences between IICC and our DS-SimCLR model is how the slices of a 3D volume are modelled: IICC models them as a sequence, while DS-SimCLR models them as a set. Assume N is the number of slices to model a scan, and C is the latent vector dimensionality representing each slice. IICC uses 1D deformable attention in its transformer, and its time complexity was shown to be $\Omega\left(N^2C + NC^2 + (k^2+2)NC\right)$, where k is the offset network dimensionality [30]. Further, we show that the Deep Set module in DS-SimCLR has time complexity $\Theta(NC + CM)$, this is described in Appendix E.2. Therefore, IICC is quadratic in the number of slices and latent dimension, whereas Deep Set is linear in these variables, making its methodology of processing slices much more efficient — in addition to more performant in classification tasks (as shown in Section 4.1).

We quantify the findings from the formal complexity analysis with profiling experiments to empirically analyse DS-SimCLR's computational burden versus the state-of-the-art IICC model. All timings were performed on a single NVIDIA Quadro RTX 8000 GPU. Reported values are an average of 4 random trials of 1000 training steps each. From Fig. 3, we see that the computational burden induced by the three distinct pretraining phases of IICC results in a significantly higher running time than the other models, even at lower sequence lengths (the sequence length used in the official implementation is 64). Lastly, the baseline SimCLR approach takes marginally longer to train than DS-SimCLR, but not significantly so ($\alpha > 0.05$). The asynchronous parallelised nature of the data loading means the effect of loading additional images for DS-SimCLR is somewhat reduced.

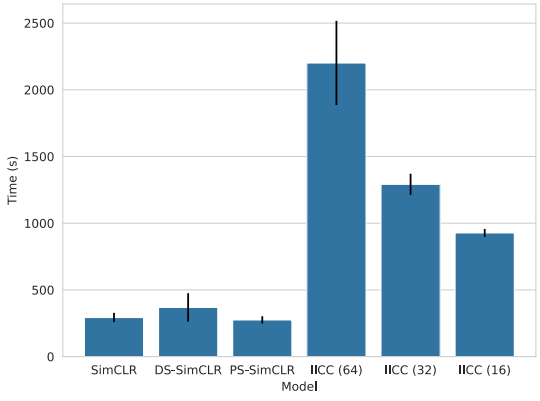


Fig. 3. Total running time for the different models. The number for each IICC method represents the sequence length used during training (i.e. the number of slices for each scan). 95% confidence intervals are shown on each bar.

5. Conclusion

We choose to model the 3D volumes as a set using the DeepSet since this architecture has proven to work well in supervised medical settings. Our integration of the DeepSet into an SS architecture (DS-SimCLR) has multiple benefits, including implicitly modelling the 3D nature of the data while notably improving performance for little additional overhead. We contrast against existing approaches which are either inherently 3D architectures or use complicated transformer-based mechanisms (such as IICC), which are significantly more computationally expensive to train than our proposed methods, and less performant in classification tasks (although more performant in dense tasks). Further, our simple sampling strategy proposed for PS-SimCLR improves upon the baseline with little additional training overhead.

CRediT authorship contribution statement

David Torpey: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Richard Klein:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2024.09.005.

References

[1] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 1597–1607.

[2] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G. Hinton, Big self-supervised models are strong semi-supervised learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.

[3] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: Proceedings of the International Conference on Computer Vision, 2021.

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

[5] L. Ericsson, H.G.R. Gouk, T.M. Hospedales, How well do self-supervised models transfer? in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 5410–5419.

[6] J. Zbontar, L. Jing, I. Misra, Y. LeCun, S. Deny, Barlow twins: Self-supervised learning via redundancy reduction, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 12310–12320.

[7] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap your own latent-a new approach to self-supervised learning, Adv. Neural Inf. Process. Syst. 33 (2020) 21271–21284.

[8] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, Adv. Neural Inf. Process. Syst. 33 (2020) 9912–9924.

[9] M. Zaheer, S. Kottur, S. Ravanbhakhsh, B. Póczos, R. Salakhutdinov, A.J. Smola, Deep sets, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

[10] S. Hemati, S. Kalra, C. Meaney, M. Babaie, A. Ghodsi, H. Tizhoosh, CNN and deep sets for end-to-end whole slide image representation learning, in: Proceedings of the Fourth Conference on Medical Imaging with Deep Learning, 2021.

[11] A. Taleb, W. Loetzsch, N. Danz, J. Severin, T. Gaertner, B. Bergner, C. Lippert, 3D self-supervised methods for medical imaging, in: Advances in Neural Information Processing Systems, 2020.

[12] Y. Tang, D. Yang, W. Li, H.R. Roth, B. Landman, D. Xu, V. Nath, A. Hatamizadeh, 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, CVPR, 2022.

[13] D.M.H. Nguyen, H. Nguyen, T.T.N. Mai, T. Cao, B.T. Nguyen, N. Ho, P. Swoboda, S. Albarqouni, P. Xie, D. Sonntag, Joint self-supervised image-volume representation learning with intra-inter contrastive clustering, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, 2023.

[14] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, A. Zisserman, With a little help from my friends: Nearest-neighbor contrastive learning of visual representations, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021.

[15] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020.

[16] C. Zhang, K. Zhang, C. Zhang, T.X. Pham, C.D. Yoo, I.S. Kweon, How does SimSiam avoid collapse without negative samples? A unified understanding with self-supervised contrastive learning, in: International Conference on Learning Representations, 2022, URL https://openreview.net/forum?id=bwq6O4Cwdl.

[17] D. Torpey, R. Klein, On the robustness of self-supervised representations for multi-view object classification, Pattern Recognit. Lett. 161 (2022) 82–89, http://dx.doi.org/10.1016/j.patrec.2022.07.016, URL https://www.sciencedirect.com/science/article/pii/S0167865522002276.

[18] L. Jing, P. Vincent, Y. LeCun, Y. Tian, Understanding dimensional collapse in contrastive self-supervised learning, in: International Conference on Learning Representations, 2022, URL https://openreview.net/forum?id=YevsQ05DEN7.

[19] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: International Conference on Learning Representations, 2018.

[20] R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: ECCV, 2016.

[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.

[22] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: ECCV, 2016.

[23] M. Kang, H. Song, S. Park, D. Yoo, S. Pereira, Benchmarking self-supervised learning on diverse pathology datasets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023.

[24] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv arXiv:1807.03748.

[25] C. Doersch, A. Gupta, A.A. Efros, Unsupervised visual representation learning by context prediction, in: IEEE International Conference on Computer Vision, 2015, pp. 1422–1430.

[26] A. Dosovitskiy, J.T. Springenberg, M. Riedmiller, T. Brox, Discriminative unsupervised feature learning with convolutional neural networks, in: Advances in Neural Information Processing Systems, 2014.

[27] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, X. Zhai, An image is worth 16 × 16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.

[28] M. Caron, P. Bojanowski, J. Mairal, A. Joulin, Unsupervised pre-training of image features on non-curated data, in: Proceedings of the International Conference on Computer Vision, ICCV, 2019.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017.

[30] Z. Xia, X. Pan, S. Song, L.E. Li, G. Huang, Vision transformer with deformable attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 4794–4803.

[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[32] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, ICLR, 2015.