CLOCS: Contrastive Learning of Cardiac Signals

Dani Kiyasseh
Department of Engineering Science
University of Oxford
Oxford, UK
dani.kiyasseh@eng.ox.ac.uk

Tingting Zhu
Department of Engineering Science
University of Oxford
Oxford, UK
tingting.zhu@eng.ox.ac.uk

David A. Clifton
Department of Engineering Science
University of Oxford
Oxford, UK
david.clifton@eng.ox.ac.uk

Abstract

The healthcare industry generates troves of unlabelled physiological data. This data can be exploited via contrastive learning, a self-supervised pre-training mechanism that encourages representations of instances to be similar to one another. We propose a family of contrastive learning methods, CLOCS, that encourages representations across time, leads, *and* patients to be similar to one another. We show that CLOCS consistently outperforms the state-of-the-art approach, SimCLR, on both linear evaluation and fine-tuning downstream tasks. We also show that CLOCS achieves strong generalization performance with only 25% of labelled training data. Furthermore, our training procedure naturally generates patient-specific representations that can be used to quantify patient-similarity.

1 Introduction

At present, the healthcare system is unable to sufficiently leverage the large, unlabelled datasets that it generates on a daily basis. This is partially due to the dependence of deep learning algorithms on high quality labels for good generalization performance. However, arriving at such high quality labels in a clinical setting where physicians are squeezed for time and attention is increasingly difficult. To overcome such an obstacle, self-supervised techniques have emerged as promising methods. These methods exploit the unlabelled dataset to formulate pretext tasks such as predicting the rotation of images Gidaris et al. [2018], their corresponding colourmap Larsson et al. [2017], and the arrow of time Wei et al. [2018]. More recently, contrastive learning was introduced as a way to learn representations across different instances that share some context. By capturing this high-level shared context (e.g. medical diagnosis), representations become invariant to the differences (e.g. input modalities) between the instances.

Contrastive learning can be characterized by three main components: 1) a positive and negative set of examples, 2) a notion of different views, and 3) a variant of the noise contrastive estimation loss. Most research in this domain has focused on curating a positive set of examples by exploiting data temporality [Oord et al., 2018], data augmentations [Chen et al., 2020], and multiple views of the same data instance [Tian et al., 2019]. These methods are predominantly catered to the image-domain and central to their implementation is the notion that shared context arises from the same instance. We believe this precludes their applicability to the medical domain where physiological time-series are plentiful. Moreover, their interpretation of shared context is limited to data from a common source where that source is the individual data instance. In medicine, however, shared context can occur

Preprint. Under review.

at a higher level, the patient level. This idea is central to our contributions and will encourage the development of representations that are patient-specific. Such representations have the potential to be used in tasks that exploit patient similarity such as disease subgroup clustering and discovery. As a result of the process, medical practitioners may receive more interpretable outputs from networks.

In this work, we leverage electrocardiogram (ECG) signals to learn patient-specific representations in a self-supervised manner via contrastive learning. To do so, we exploit the fact that ECG signals summarize both temporal and spatial information. The latter can be understood in terms of multiple lead measurements which represent the projection of an electrical signal onto multiple axes (see Fig. 1).

Contributions. Our contributions are as following:

1. **Contrastive Multi-Segment Coding** - we encourage representations of pairs of temporal segments of data from the same patient to be similar to one another.
2. **Contrastive Multi-Lead Coding** - we encourage representations of multiple views (electrocardiogram leads) of the same temporal segment from the same patient to be similar to one another.
3. **Contrastive Multi-Segment and Multi-Lead Coding** - we encourage representations of both temporal segments and multiple leads from the same patient to be similar to one another.
4. **Patient-Specific Noise Contrastive Estimation Loss** - we adapt the loss introduced by Tian et al. [2019] to account for patient-specific instances.

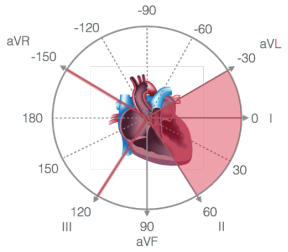


Figure 1: Hexiaxial lead diagram illustrating the angles of the various leads [Michael, 2014]. Each lead measurement represents the projection of the same electrical signal onto a different axis.

2 Related Work

Contrastive Learning. In contrastive predictive coding, Oord et al. [2018] use representations of current segments to predict those of future segments. More recently, Tian et al. [2019] propose contrastive multi-view coding where multiple views of the same image are treated as 'shared context'. With each view given a unique encoder, their approach does not scale as the number of encoders is $\mathcal{O}(N)$ where N is the number of views. Chen et al. [2020], He et al. [2019] exploit the idea of instance discrimination [Wu et al., 2018] and propose that multiple views are stochastically augmented forms of the same instance. They explore the benefit of various sequences of data augmentation compositions and show that cropping and colour distortions are the most important. These augmentations do not easily extend to the time-series domain. Shen et al. [2020] consider changing the multiple views component, proposing to create mixtures of images to smoothen the output distribution and thus prevent the model from being overly confident. Time Contrastive Learning [Hyvarinen and Morioka, 2016] performs contrastive learning over temporal segments in a signal and illustrate the relationship between their approach and ICA. In contrast to our work, they formulate their task as prediction of the segment index within a signal with limited experiments that do not exploit the noise contrastive estimation (NCE) loss. Bachman et al. [2019] propose AMDIM which focuses on maximizing the similarity between global and local features in the network. This approach involves a number of tricks to supposedly decrease the instability of the NCE loss e.g., regularization of matching scores, and clipping the scores. They also apply data augmentation to images, yet do not explore its extension to medical time-series. Time Contrastive Networks [Sermanet et al., 2017] attempt to learn commonalities across views and differences across time. In contrast, our work focuses on identifying commonalities across *both* views and time.

Self-Supervision for Medical Time-Series. Miotto et al. [2016] propose DeepPatient, a 3-layer stacked denoising autoencoder that attempts to learn a patient representation using electronic health record (EHR) data. Although performed on a large proprietary dataset, their approach is focused on EHRs and does not explore contrastive learning for physiological signals. Sarkar and Etemad [2020] apply existing self-supervised methods on ECG recordings in the context of affective computing. The methods implemented include defining pretext classification tasks such as temporal inversion, negation, time-warping, etc. Minimal differences are observed between these methods. Their work is

limited to affective computing, does not explore contrastive learning, and does not exploit multi-lead data as we do. Lyu et al. [2018] explore a sequence to sequence model to learn representations from EHR data in the eICU dataset. They minimize the reconstruction error of the input time-series. Li et al. [2020] leverage the aforementioned unsupervised learning technique on a large clinical dataset, CPRD, to obtain uncertainty estimates for predictions.

Patient Similarity. Pai and Bader [2018] provide an overview of work focused on patient similarity and introduce the Patient Similarity Network (PSN) framework where each node in a graph is represented by a patient. PSNs can be used to assist in the classification of particular diseases or to cluster patients according to disease subgroups. Pai et al. [2019] exploit this framework to model cancer survival in an interpretable manner while illustrating its superiority relative to traditional machine learning methods. Darabi et al. [2019] implement a multi-stage unsupervised learning pipeline that consists of training an autoencoder, a transformer, and a network tasked with predicting next visit medications, diagnoses, and treatments. The representation learned from this process is used for mortality and readmission prediction. They name this representation the Patient Status Vector (PSV). This is similar in spirit to the notion of Medical Concept Embedding [Cai et al., 2018]. Very few of these methods rigorously validate the reliability of their patient representations beyond simply using them for downstream clinical tasks. A byproduct of our work is the learning of a patient-specific representation - one that can sufficiently summarize the cardiac state of a patient. We attempt to do so in a self-supervised manner using contrastive learning.

3 Background

3.1 Contrastive Learning

Assume the presence of a learner f_θ, parameterized by θ, that is exposed to unlabelled instances $x \in X^{N \text{x} D}$. Further assume a pair of representations, $h_A^i = f_\theta(x_A^i)$ and $h_B^k = f_\theta(x_B^k)$ where $i, k \in N$, that belong to two different views, A and B. Such views can be transformations, T, applied to each instance where $x_A^i = T_A(x^i)$ and $x_B^k = T_B(x^k)$. For instance, these transformations can convert images into their grayscale and depth-map counterparts or simply consist of two different data augmentation procedures such as random cropping and flipping.

In contrastive learning, representations with shared context (positive pairs) are encouraged to be similar to one another and dissimilar to the remaining representations (negative pairs). Typically, shared context is interpreted as representations belonging to the same instance, i.e., $i = k$. To quantify the similarity of these representations, a similarity matrix is constructed (see Fig. 2) where the pairwise distance, $s(h_A^i, h_B^k)$, for some similarity metric, s, between each representation from views A and B is calculated. This matrix is set up such that its diagonal entries are positive pairs and its off-diagonals are negative pairs. By encouraging high similarity between representations in the positive pair, the goal is to learn representations that are invariant to different views of the same instance.

4 Methods

4.1 Definition of Positive and Negative Pairs

Shared context typically assumes that representations belong to the same instance. Inherent to our approach, however, is the redefinition of shared context to refer to representations that belong to the same *patient*. In our similarity matrix setup (see Fig. 2), this approach implies that positive pairs constitute not only diagonal entries (green) but also select off-diagonals (yellow) that belong to the same patient.

4.2 Definition of Different Views

Representations with a shared context are encouraged to be similar across views. Consequently, the principal desideratum for choosing the views is that they should introduce some diversity to the dataset such that the network is made robust to them. To this end, we propose views that exploit the temporal and spatial information present within an ECG.

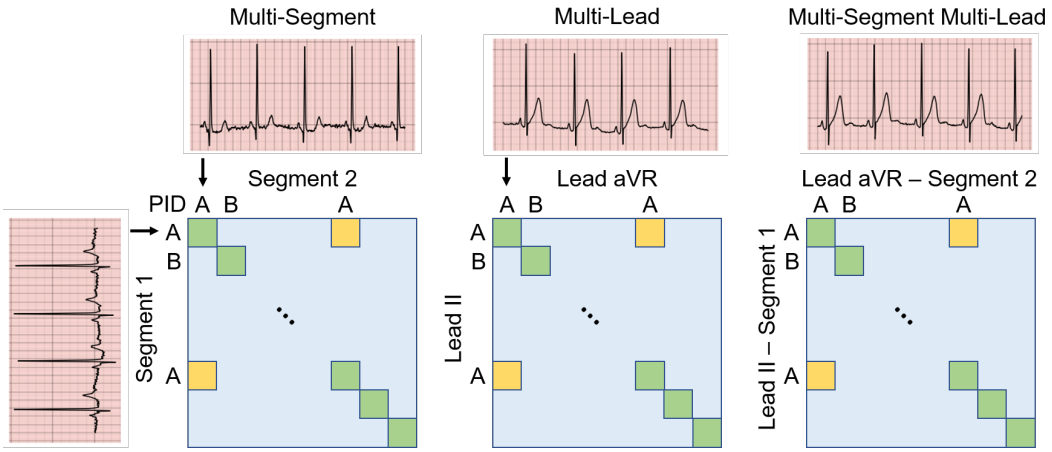


Figure 2: Visualization of the pairwise similarity matrices used in our three contrastive methods, (left) **Contrastive Multi-Segment Coding**, (center) **Contrastive Multi-Lead Coding**, and (right) **Contrastive Multi-Segment Multi-Lead Coding**. Additional matrices would be generated based on all possible pairs of views. Unlike traditional contrastive learning which only considers diagonal components (green boxes) as positive pairs, our work also considers off-diagonal components which correspond to instances that share the same patient ID (PID) (yellow boxes). Blue area corresponds to negative examples as they pertain to instances from different patients.

Contrastive Multi-Segment Coding (CMSC). In this setting, adjacent and non-overlapping temporal segments of a time-series signal are considered to be a positive pair (see Fig. 2 left). Formally, $A = t_1$ and $B = t_2$ where t indicates the timestamp of the temporal segment.

Contrastive Multi-Lead Coding (CMLC). In this setting, different projections of the same electrical signal i.e., leads, that correspond to the same temporal segment are considered to be positive pairs (see Fig. 2 center). Formally, $A = L_1$ and $B = L_2$ where $L \in (\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{aVR}, \mathrm{aVL}, \mathrm{aVF}, \mathrm{V1}, \mathrm{V2}, \mathrm{V3}, \mathrm{V4}, \mathrm{V5}, \mathrm{V6})$ is the set of projections. Since the similarity matrix is computed for each pair of views, the number of similarity matrices constructed in this setting is equal to $\binom{|L|}{2}$ where $|L|$ is equal to the total number of projections used.

Contrastive Multi-Segment Multi-Lead Coding (CMSMLC). In this setting, we combine the two previous approaches such that different projections regardless of whether they correspond to the same temporal segment are considered to be a positive pair (see Fig. 2 right). Formally, $A = L_{t_1}$ and $B = L_{t_2}$ where L and t correspond to the projection and timestamp of the temporal segment, respectively. In this setting, the number of similarity matrices constructed is equal to $2 \cdot \binom{|L|}{2}$.

4.3 Variant of Noise Contrastive Estimation Loss

The noise contrastive estimation loss encourages representations in the positive pair to be similar to one another and dissimilar to the remaining representations. Given our patient-centric definition of positive pairs, we encourage representations, h_A^i and h_B^k from a pair of views where $i, k \in P$, from the same patient, P, to be similar to one another. Within each mini-batch, the similarity of these positive pair representations include both the diagonal and potentially off-diagonal entries in the similarity matrices in Fig. 2. To construct these matrices, we use the cosine similarity, s, with a temperature scaling parameter, τ, as is performed in [Tian et al., 2019, Chen et al., 2020]. Mathematically, our objective function for all combinations of pairs of views, A and B, is as follows:

$$\mathcal{L} = \mathbb{E}_{A,B}\left[\mathcal{L}^{h_A,h_B}_{diagonal} + \mathcal{L}^{h_B,h_A}_{diagonal} + \mathcal{L}^{h_A,h_B}_{off-diagonal} + \mathcal{L}^{h_B,h_A}_{off-diagonal}\right] \tag{1}$$

$$\mathcal{L}^{h_A,h_B}_{diagonal} = \mathbb{E}_{i \in P}\left[\frac{e^{s(h_A^i, h_B^i)}}{\sum_j e^{s(h_A^i, h_B^j)}}\right] \tag{2}$$

$$\mathcal{L}_{off-diagonal}^{h_A,h_B} = \mathbb{E}_{i,k\in P}\left[\frac{e^{s(h_A^i,h_B^k)}}{\sum_j e^{s(h_A^i,h_B^j)}}\right] \tag{3}$$

$$s(h_A^i, h_B^i) = \frac{f_\theta(x_A^i)\cdot f_\theta(x_B^i)}{\|f_\theta(x_A^i)\|\|f_\theta(x_B^i)\|}\frac{1}{\tau} \tag{4}$$

5 Experimental Design

5.1 Datasets

We conduct our experiments on four ECG datasets that include cardiac arrhythmia labels. **PhysioNet 2020** consists of 12-Lead ECG recordings from 6,877 patients alongside 9 different classes of cardiac arrhythmia. Each recording can be associated with multiple labels. **Chapman** [Zheng et al., 2020] consists of 12-Lead ECG recordings from 10,646 patients alongside 11 different classes of cardiac arrhythmia. As is suggested by Zheng et al. [2020], we group these labels into 4 major classes. **PhysioNet 2017** [Clifford et al., 2017] consists of 8,528 single-lead ECG recordings alongside 4 different classes. Lastly, **Cardiology** [Hannun et al., 2019] consists of single-lead ECG recordings from 328 patients alongside 12 different classes of cardiac arrhythmia. An in-depth description of these datasets can be found in Appendix A.1.

All datasets were split into training, validation, and test sets according to patient ID using a 60, 20, 20 configuration. In other words, patients appeared in only one of the sets. The exact number of instances used during self-supervised pre-training and supervised training can be found in Appendix A.2.

5.2 Pre-Training Implementation

We conduct our pre-training experiments on the training set of two of the four datasets: PhysioNet 2020 and Chapman. We chose these datasets as they contain multi-lead data. The implementation details of the various pre-training methods are as follows. In **CMSC**, we extract a pair of non-overlapping temporal segments of $S = 2500$ samples. Therefore, our model is presented with a mini-batch of dimension $B\text{x}S\text{x}2$ where B is the batchsize, and S is the number of samples. In **CMLC**, we explore two scenarios with different numbers of leads corresponding to the same instance. Our mini-batch dimension is $B\text{x}S\text{x}L$, where L is the number of leads. Laslty, in **CMSMLC**, we incorporate an additional temporal segment in each mini-batch. Therefore, our mini-batch dimension is $B\text{x}2S\text{x}L$.

To ensure a fair comparison between all methods, we expose them to an equal number of patients and instances during training. In CMLC or CMSMLC, we either pre-train using 4 leads (II, V2, aVL, aVR) or all 12 leads. We chose these 4 leads as they cover a large range of axes, as shown in Fig. 1.

5.3 Evaluation on Downstream Task

We evaluate our pre-trained methods in two scenarios. In **Linear Evaluation of Representations**, we are interested in evaluating the utility of the fixed feature extractor in learning representations. Therefore, the pre-trained parameters are frozen and multinomial logistic regression is performed on the downstream supervised task. In **Transfer Capabilities of Representations**, we are interested in evaluating the inductive bias introduced by pre-training. Therefore, the pre-trained parameters are used as an initialization for training on the downstream supervised task.

5.4 Baselines

We compare our pre-training methods to the state-of-the-art approach, **SimCLR**. We adapt SimCLR [Chen et al., 2020] to the time-series domain where augmentations are introduced as additive Gaussian perturbations, $\epsilon \sim \mathcal{N}(0, \sigma)$ and σ was chosen based on the amplitude of the signal. In this setting, instances and their perturbed counterparts are encouraged to have similar representations. When evaluating the transfer capabilities of representations, we compare parameters initialized based on our methods to those based on a random initialization, **Random**.

5.5 Hyperparameters

During self-supervised pre-training, we set the temperature parameter in the objective function, $\tau = 0.06$, as per [Chen et al., 2020]. Experiments were implemented in PyTorch [Paszke et al., 2019] using the Adam optimizer with a batchsize of 256 and a learning rate of 10^{-4}. The network architecture can be found in Appendix B.

6 Experimental Results

6.1 Linear Evaluation of Representations

In this section, we evaluate the utility of the self-supervised representations learned using four leads on a downstream linear classification task. In Table 1, we show the test AUC on Chapman and PhysioNet 2020 using 50% of the labelled data ($F = 0.5$) after having learned representations, with dimension $E = 128$, using these two datasets.

We show that CMSC outperforms the state-of-the-art method, SimCLR, on both datasets. On the Chapman dataset, these two methods achieve an $\mathrm{AUC} = 0.896$ and 0.738, respectively, a 15.8% improvement. Such a finding implies that the representations learned with CMSC are richer and thus allow for improved generalization. We hypothesize that this is due to the setup of CMSC whereby the shared context is across segments (temporally) and patients. In Appendix C, we extend these findings to show that CLOCS outperforms SimCLR in 100% of all experiments conducted. Furthermore, we show that these conclusions also hold when pre-training and evaluating using all 12 leads (see Appendix C.2).

Table 1: Test AUC of the linear evaluation of the representations at $F = 0.5$, after having pre-trained on Chapman or PhysioNet 2020 with $E = 128$. Pre-training and evaluating multi-lead datasets* using 4 leads (II, V2, aVL, aVR). Mean and standard deviation are shown across 5 seeds.

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.738 ± 0.034	0.615 ± 0.014
CMSC	$\mathbf{0.896 \pm 0.005}$	$\mathbf{0.715 \pm 0.033}$
CMLC	0.870 ± 0.022	0.596 ± 0.008
CMSMLC	0.847 ± 0.024	0.680 ± 0.008

6.2 Transfer Capabilities of Representations

In this section, we evaluate the utility of initializing a network for a downstream task with parameters learned via self-supervision using four leads. In Table 2, we show the test AUC on downstream datasets at $F = 0.5$ for the various self-supervised methods with $E = 128$. The rest of the results can be found in Appendix D.

We show that, with a few exceptions, self-supervision is advantageous relative to a random initialization. This can be seen by the higher AUC achieved by the former relative to the latter. We also show that CMSC or CMSMLC outperform SimCLR, depending on the downstream dataset. For instance, when pre-training on Chapman and fine-tuning on Cardiology, CMSMLC achieves an $\mathrm{AUC} = 0.717$, a 3.9% improvement compared to SimCLR. This implies that by encouraging representations across time, leads, and patients to be similar to one another, networks are nudged into a favourable parameter space. In Appendix D.1, we extend these findings and illustrate that CLOCS outperforms SimCLR in at least 75% of all experiments conducted, on average. When pre-training, fine-tuning, and evaluating using all 12 leads, we show that CMSC outperforms all other methods in at least 90% of all experiments conducted (see Appendix D.2).

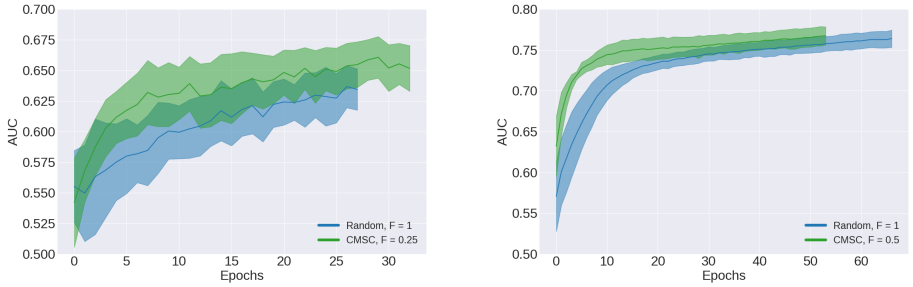
Table 2: Test AUC in the fine-tuning scenario at $F = 0.5$, after having pre-trained on Chapman or PhysioNet 2020 with $E = 128$. Pre-training, fine-tuning, and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.678 ± 0.011	0.763 ± 0.005	0.803 ± 0.008	0.678 ± 0.011	0.763 ± 0.005	0.907 ± 0.006
SimCLR	0.676 ± 0.011	0.772 ± 0.010	0.823 ± 0.011	0.658 ± 0.027	0.762 ± 0.009	0.923 ± 0.010
CMSC	0.695 ± 0.024	0.773 ± 0.013	$\mathbf{0.830 \pm 0.002}$	$\mathbf{0.714 \pm 0.014}$	0.7600 ± 0.013	$\mathbf{0.932 \pm 0.008}$
CMLC	0.665 ± 0.016	0.767 ± 0.013	0.810 ± 0.011	0.675 ± 0.013	0.762 ± 0.007	0.910 ± 0.012
CMSMLC	$\mathbf{0.717 \pm 0.006}$	$\mathbf{0.774 \pm 0.004}$	0.814 ± 0.009	0.698 ± 0.011	$\mathbf{0.774 \pm 0.012}$	0.930 ± 0.012

6.3 Doing More With Less Labelled Data

As self-supervision can nudge the network to a favourable parameter space, we set out to investigate whether such a space can lead to strong generalization with less labelled data in the downstream task. In Fig. 3, we illustrate the validation AUC of two datasets after having started training with a random initialization or one from CMSC pre-training.

Training with a CMSC initialization drastically improves sample-efficiency. In Fig. 3a, we show that training with a CMSC initialization and being exposed to only 25% of the labelled data outperforms that with a random initialization that is exposed to 100% of the labelled data. This can be seen by the consistently higher AUC during and at the end of training. A similar outcome can be seen in Fig. 3b. This suggests that self-supervised pre-training exploits data efficiently such that it can do more with less on downstream classification tasks.

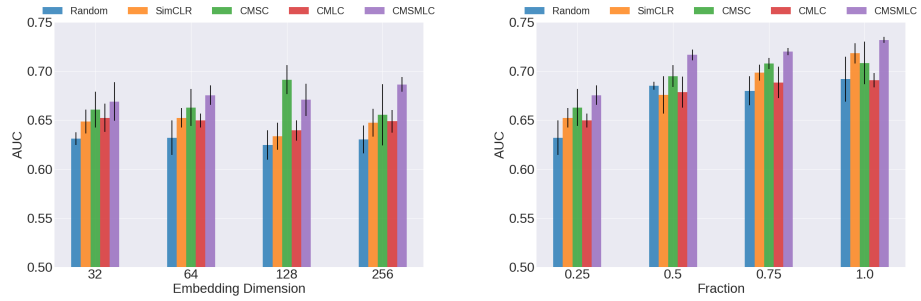


(a) PhysioNet 2020 → Cardiology (b) Chapman → PhysioNet 2017

Figure 3: Validation AUC starting with a random and self-supervised initialization exposed to different amounts of labelled training data. Results are averaged across 5 seeds. Shaded area represents one standard deviation.

6.4 Effect of Embedding Dimension, *E*, and Availability of Labelled Data, *F*

The dimension of the representation learned during self-supervision and the availability of labelled training data can both have an effect on model performance. We investigate these claims in Fig, 4. We illustrate the test AUC for all pre-training methods as a function of $E = (32, 64, 128, 256)$ and $F = (0.25, 0.50, 0.75, 1)$ in Figs. 4a and 4b, respectively.



(a) Chapman → Cardiology, $F = 0.25$ (b) Chapman → Cardiology, $E = 64$

Figure 4: Effect of (a) embedding dimension and (b) labelled fraction on the test AUC when transferring from Chapman to Cardiology. Results are averaged across 5 seeds. Error bars represent one standard deviation.

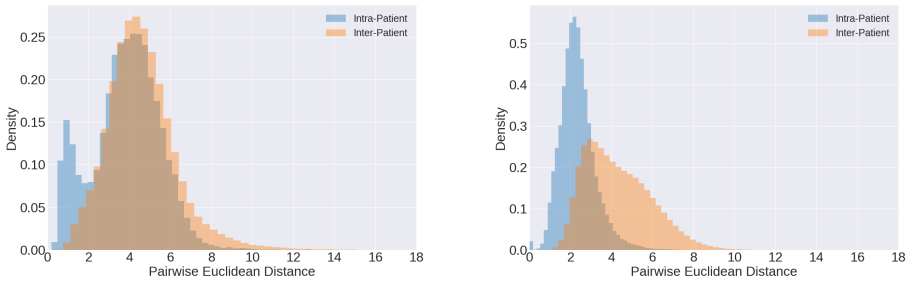
In Fig. 4a, we show that training procedures with random initializations and those based on SimCLR are not significantly affected by the embedding dimension. This can be seen by the $\text{AUC} \approx 0.63$ and ≈ 0.65, for these two methods across all values of E. In contrast, the embedding dimension has a greater effect on CMSC where $\text{AUC} \approx 0.66 \rightarrow 0.69$ as $E = 32 \rightarrow 128$. This implies that CMSC is still capable of achieving strong generalization performance despite the presence of few labelled data ($F = 0.25$).

In Fig. 4b, we show that increasing the amount of labelled training data positively benefits the generalization performance of all methods. This can be seen by the increasing AUC values as $F = 0.25 \rightarrow 1$. We also show that at all fraction values, CMSMLC outperforms its counterparts. With 100% of the labelled data, CMSMLC achieves an $\text{AUC} = 0.732$ whereas SimCLR achieves an $\text{AUC} = 0.718$. Such superiority still holds with 25% of the labelled data where the two methods achieve an $\text{AUC} = 0.675$ and 0.652, respectively. This outcome emphasizes the robustness of CMSMLC to scarce labelled training data.

6.5 Patient-Specific Representations

Inherent to our contrastive approach is the redefinition of 'shared context' to refer to instances from the same patient. In other words, instances should have more similar representations if they belong to the same patient. To determine whether this was satisfied, we calculate the pairwise Euclidean distance between intra-patient instances and inter-patient instances. The former should be, on average, smaller than the latter. In Fig. 5, we illustrate the two distributions associated with the intra and inter-patient distances at $E = 128$. We also find that increasing the embedding dimension shifts these distributions to higher values (see Appendix E).

We show that these two distributions have large mean values and overlap significantly when implementing SimCLR, as seen in Fig. 5a. This is expected as the SimCLR is blind to the notion of a patient. In contrast, when implementing CMSC, the intra-patient distances are lower than those found in SimCLR, as seen in Fig. 5b. Moreover, the intra and inter-patient distributions are more separable. This implies that CMSC leads to instances from the same patient to share similar representations, and to those from different patients to have dissimilar representations. We note that this distinction takes place while learning better representations as observed in previous sections.



(a) SimCLR (b) CMSC

Figure 5: Distribution of pairwise Euclidean distance between representations ($E = 128$) belonging to the same patient and those belonging to different patients. Self-supervision was performed on PhysioNet 2020. Notice the lower average intra-patient values and improved separability with CMSC.

7 Discussion and Future Work

In this paper, we proposed a family of self-supervised pre-training mechanisms, entitled CLOCS, based on contrastive learning for physiological signals. In the process, we encourage representations across segments (temporally) and leads (spatially) that correspond to instances from the same patient to be similar to one another. We show that our methods outperform the state-of-the-art method, SimCLR, when performing a linear evaluation of and fine-tuning on downstream tasks. This conclusion also holds when pre-training and evaluating with a different number of leads. We also illustrate the ability of our methods to learn patient-specific representations. We now elucidate several future avenues worth exploring.

Quantifying Patient-Similarity. We have managed to learn patient-specific representations. These representations can be used to quantify patient-similarity in order to assist with diagnosis or gain a better understanding of a diseased condition. Validation of these representations can be performed by comparing known similar patients.

Multi-Modal Transfer. We transferred parameters from one task to another that shared the same input modality, the ECG. Such data may not always be available for self-supervision. An interesting

path would be to explore whether contrastive self-supervision on one modality can transfer well to another modality.

8 Broader Impact

With the healthcare industry becoming increasingly digitized, the amount of data that is generated is growing precipitously. Due to the scale and rate with which this growth is taking place, annotating all this data, to feed into a deep-learning algorithm for instance, is infeasible. Through contrastive learning, we can exploit this unlabelled data to learn representations that improve generalization in scenarios where labelled data is scarce. Such scenarios are prevalent in low-resource clinical settings where the data-recording infrastructure is absent and medical conditions are rare and different to those experienced elsewhere. Our work can contribute to this domain of transfer learning, arguably the hallmark of machine learning for healthcare, in order to improve clinical decision support systems and patient outcomes.

Although we showed that CLOCS outperforms a random initialization, recent work has cast doubt on the utility of *supervised* pre-training when fine-tuning on medical datasets [Raghu et al., 2019]. Similarly, our work is limited in that it transfers information between tasks that share the same input modality, the ECG. Its applicability to areas where input distribution shift exists requires further research. Nonetheless, we believe that the risks associated with CLOCS are minimal given that it is a pre-training method that can be overwritten.

References

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017.

Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Sarah Michael. Senior report 8.8, 2014. URL `https://drhem.wordpress.com/tag/cardiovascular/`. [Online; accessed March 30, 2020].

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, and Trevor Darrell. Rethinking image mixture for unsupervised visual representation learning. *arXiv preprint arXiv:2003.05438*, 2020.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. Time-contrastive networks: Self-supervised learning from multi-view observation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 486–487. IEEE, 2017.

Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, 6(1): 1–10, 2016.

Pritam Sarkar and Ali Etemad. Self-supervised ecg representation learning for emotion recognition. *arXiv preprint arXiv:2002.03898*, 2020.

Xinrui Lyu, Matthias Hueser, Stephanie L Hyland, George Zerveas, and Gunnar Rätsch. Improving clinical predictions through unsupervised time series representation learning. *arXiv preprint arXiv:1812.00490*, 2018.

Yikuan Li, Shishir Rao, Abdelaali Hassaine, Rema Ramakrishnan, Yajie Zhu, Dexter Canoy, Gholamreza Salimi-Khorshidi, Thomas Lukasiewicz, and Kazem Rahimi. Deep bayesian gaussian processes for uncertainty estimation in electronic health records. *arXiv preprint arXiv:2003.10170*, 2020.

Shraddha Pai and Gary D Bader. Patient similarity networks for precision medicine. *Journal of Molecular Biology*, 430(18):2924–2938, 2018.

Shraddha Pai, Shirley Hui, Ruth Isserlin, Muhammad A Shah, Hussam Kaka, and Gary D Bader. netdx: Interpretable patient classification using integrated patient similarity networks. *Molecular Systems Biology*, 15(3), 2019.

Sajad Darabi, Mohammad Kachuee, and Majid Sarrafzadeh. Unsupervised representation for ehr signals and codes as patient status vector. *arXiv preprint arXiv:1910.01803*, 2019.

Xiangrui Cai, Jinyang Gao, Kee Yuan Ngiam, Beng Chin Ooi, Ying Zhang, and Xiaojie Yuan. Medical concept embedding with time-aware attention. *arXiv preprint arXiv:1806.02873*, 2018.

Jianwei Zheng, Jianming Zhang, Sidy Danioko, Hai Yao, Hangyuan Guo, and Cyril Rakovski. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific Data*, 7(1):1–8, 2020.

Gari D Clifford, Chengyu Liu, Benjamin Moody, H Lehman Li-wei, Ikaro Silva, Qiao Li, AE Johnson, and Roger G Mark. Af classification from a short single lead ECG recording: the physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology*, pages 1–4, 2017.

Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25(1):65, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems*, pages 3342–3352, 2019.

A Datasets

A.1 Data Preprocessing

For all of the datasets, frames consisted of 2500 samples and consecutive frames had no overlap with one another. Data splits were always performed at the patient-level.

PhysioNet 2020. Each ECG recording varied in duration from 6 seconds to 60 seconds with a sampling rate of 500Hz. Each ECG frame in our setup consisted of 2500 samples (5 seconds). We assign multiple labels to each ECG recording as provided by the original authors. These labels are: AF, I-AVB, LBBB, Normal, PAC, PVC, RBBB, STD, and STE. The ECG frames were normalized in amplitude between the values of 0 and 1.

Chapman. Each ECG recording was originally 10 seconds with a sampling rate of 500Hz. We downsample the recording to 250Hz and therefore each ECG frame in our setup consisted of 2500 samples. We follow the labelling setup suggested by Zheng et al. [2020] which resulted in four classes: Atrial Fibrillation, GSVT, Sudden Bradychardia, Sinus Rhythm. The ECG frames were normalized in amplitude between the values of 0 and 1.

Cardiology. Each ECG recording was originally 30 seconds with a sampling rate of 200Hz. Each ECG frame in our setup consisted of 256 samples resampled to 2500 samples. Labels made by a group of physicians were used to assign classes to each ECG frame depending on whether that label coincided in time with the ECG frame. These labels are: AFIB, AVB, BIGEMINY, EAR, IVR, JUNCTIONAL, NOISE, NSR, SVT, TRIGEMINY, VT, and WENCKEBACH. Sudden bradycardia cases were excluded from the data as they were not included in the original formulation by the authors. The ECG frames were not normalized.

PhysioNet 2017. Each ECG recording originally varied in length between 9 and 30 seconds with a sampling rate of 300Hz. Each ECG frames in our setup consisted of 2500 samples. We use the original labels, resulting in four classes: Normal, AF, Other, and Noisy. The ECG frames were not normalized.

A.2 Data Samples

A.2.1 Self-supervised Pre-training

In this section, we outline the dimension of the inputs used for the various pre-training methods. They are expressed in the form of NxSxL where N is the total number of instances, S is the frame length of each instance, and L (if applicable) is the number of leads used. Where L is not explicitly mentioned, we report values with four leads as this was primarily used for all experiments conducted.

Table 3: Dimension of the input data, NxSxL, used during the training and validation phases of the various self-supervised pre-training methods. $S = 2500$ is the number of samples in each instance fed to the network. L is the number of leads (projections) used during pre-training.

Dataset	Method	Train	Validation
PhysioNet 2020	SimCLR	51,880 x S	12,948 x S
	CMSC	24,080 x 2S	6,076 x 2S
	CMLC	24,080 x S x L	6,076 x S x L
	CMSMLC	6,020 x 2S x L	1,519 x 2S x L
Chapman	SimCLR	25,543 x S	8,512 x S
	CMSC	25,543 x 2S	8,512 x 2S
	CMLC	25,543 x S x L	8,512 x S x L
	CMSMLC	6,382 x 2S x L	2125 x 2S x L

A.2.2 Supervised Training

In this section, we outline the number of instances used during supervised training on the downstream tasks. For multi-lead datasets, we report these values having used four leads. A simple multiplicative factor can be used to deduce the number of instances used with a different number of leads.

Table 4: Number of instances (number of patients) used during the supervised training of the downstream tasks. For multi-lead datasets*, these represent sample sizes for the four leads (II, V2, aVL, aVR).

Dataset	Train	Validation	Test
PhysioNet 2020*	51,880 (4,402)	12,948 (1,100)	15,820 (1,375)
Chapman*	25,543 (6,387)	8,512 (2,129)	8,520 (2,130)
Cardiology	4,584 (201)	1,109 (50)	1,386 (62)
PhysioNet 2017	18,256 (5,459)	4,581 (1,364)	5,824 (1,705)

A.3 Experiment Details

Table 5: Batchsize and learning rates used for training with different datasets. The Adam optimizer was used for all experiments.

Dataset	Batchsize	Learning Rate
PhysioNet 2020	256	10^{-4}
Chapman	256	10^{-4}
Cardiology	16	10^{-4}
PhysioNet 2017	256	10^{-4}

B Network

In this section, we outline the architecture of the neural network used for all experiments. For pre-training, the final layer (Layer 5) was removed and representations with dimension E were learned. During training on the downstream tasks, the final layer was introduced.

Table 6: Network architecture used for all experiments. K, C_{in}, and C_{out} represent the kernel size, number of input channels, and number of output channels, respectively. A stride of 3 was used for all convolutional layers. E represents the dimension of the final representation.

Layer Number	Layer Components	Kernel Dimension
1	Conv 1D BatchNorm ReLU MaxPool(2) Dropout(0.1)	7 x 1 x 4 (K x C_{in} x C_{out})
2	Conv 1D BatchNorm ReLU MaxPool(2) Dropout(0.1)	7 x 4 x 16
3	Conv 1D BatchNorm ReLU MaxPool(2) Dropout(0.1)	7 x 16 x 32
4	Linear ReLU	320 x E
5	Linear	E x C (classes)

C Linear Evaluation of Representations

In this section, we evaluate the utility of the representations learned as a result of self-supervised pre-training. We pre-train on two different datasets, freeze the network parameters, and transfer them to a downstream task whereby a linear multinomial logistic regression (MLR) model is trained. In doing so, we are evaluating the richness of the representations learned. We perform these experiments under two scenarios. The first involves pre-training and evaluating using 4 leads (II, V2, aVL, aVR) (see Sec. C.1). The second involves pre-training and evaluating using all 12 leads (see Sec. C.2). We chose these two scenarios to help determine whether our findings generalize to domains where a different number of leads is available.

C.1 Pre-training and Evaluating using 4 leads

We present Tables 7 - 10 which illustrate the test AUC of an MLR evaluated on Chapman and PhysioNet 2020 after having pre-trained on these two datasets using only 4 of the 12 leads, respectively, These are presented for a range of embedding dimensions, $E = (32, 64, 128, 256)$, and available labelled training data, $F = (0.25, 0.50, 0.75, 1)$.

C.1.1 Embedding Dimension, $E = 32$

We show that CMSMLC outperforms all other methods when evaluating on Chapman, regardless of the available labelled training data. This can be seen by the higher AUC achieved by this method relative to the remaining methods. For instance, at $F = 0.25$, CMSMLC achieves an $\mathrm{AUC} = 0.844$ compared to 0.665 for SimCLR. When evaluating on PhysioNet 2020, we find that CMSC consistently outperforms the remaining methods, as seen by its higher test AUC values.

Table 7: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.665 ± 0.014	0.564 ± 0.009
CMSC	0.831 ± 0.131	**0.701 ± 0.046**
CMLC	0.789 ± 0.020	0.563 ± 0.008
CMSMLC	**0.844 ± 0.023**	0.619 ± 0.019

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.666 ± 0.015	0.587 ± 0.009
CMSC	0.831 ± 0.131	**0.707 ± 0.038**
CMLC	0.801 ± 0.016	0.572 ± 0.008
CMSMLC	**0.850 ± 0.022**	0.636 ± 0.020

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.670 ± 0.013	0.591 ± 0.010
CMSC	0.833 ± 0.129	**0.709 ± 0.039**
CMLC	0.805 ± 0.018	0.585 ± 0.009
CMSMLC	**0.850 ± 0.021**	0.643 ± 0.020

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.670 ± 0.013	0.594 ± 0.010
CMSC	0.831 ± 0.131	**0.709 ± 0.038**
CMLC	0.807 ± 0.017	0.593 ± 0.009
CMSMLC	**0.852 ± 0.021**	0.645 ± 0.021

C.1.2 Embedding Dimension, $E = 64$

We find that the conclusions arrived at with $E = 32$ are similar to those in this scenario. Namely, CMSMLC outperforms all remaining methods when evaluating on Chapman. On the other hand, CMSC outperforms all methods when evaluating on PhysioNet 2020. This can be seen by the bold test AUC values in Table 8.

Table 8: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.709 ± 0.019	0.574 ± 0.005
CMSC	0.829 ± 0.130	$\mathbf{0.720 \pm 0.012}$
CMLC	0.842 ± 0.020	0.592 ± 0.019
CMSMLC	$\mathbf{0.856 \pm 0.022}$	0.641 ± 0.023

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.722 ± 0.025	0.599 ± 0.010
CMSC	0.830 ± 0.132	$\mathbf{0.721 \pm 0.013}$
CMLC	0.850 ± 0.02	0.607 ± 0.018
CMSMLC	$\mathbf{0.861 \pm 0.02}$	0.662 ± 0.020

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.726 ± 0.023	0.604 ± 0.010
CMSC	0.831 ± 0.126	$\mathbf{0.725 \pm 0.009}$
CMLC	0.854 ± 0.021	0.619 ± 0.017
CMSMLC	$\mathbf{0.861 \pm 0.021}$	0.671 ± 0.018

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.727 ± 0.025	0.608 ± 0.010
CMSC	0.832 ± 0.126	$\mathbf{0.726 \pm 0.009}$
CMLC	0.855 ± 0.020	0.627 ± 0.015
CMSMLC	$\mathbf{0.862 \pm 0.020}$	0.673 ± 0.018

C.1.3 Embedding Dimension = 128

In this scenario and in contrast to conclusions arrived at with $E = 32$ and 64, we find that CMSC outperforms all methods when evaluated on both datasets, Chapman and PhysioNet 2020. This can be seen by the bold test AUC values in Table 13. For instance, at $F = 0.25$, CMSC achieves an $\text{AUC} = 0.895$ compared to 0.727 achieved by SimCLR. That is a 16.8% improvement relative to the state-of-the-art.

Table 9: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.727 ± 0.032	0.585 ± 0.016
CMSC	**0.895 ± 0.004**	**0.713 ± 0.032**
CMLC	0.863 ± 0.026	0.580 ± 0.007
CMSMLC	0.842 ± 0.021	0.661 ± 0.010

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.738 ± 0.034	0.615 ± 0.014
CMSC	**0.896 ± 0.005**	**0.715 ± 0.033**
CMLC	0.870 ± 0.022	0.596 ± 0.008
CMSMLC	0.847 ± 0.024	0.680 ± 0.008

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.742 ± 0.033	0.620 ± 0.015
CMSC	**0.898 ± 0.002**	**0.717 ± 0.033**
CMLC	0.872 ± 0.022	0.606 ± 0.008
CMSMLC	0.848 ± 0.023	0.685 ± 0.008

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.742 ± 0.033	0.623 ± 0.014
CMSC	**0.897 ± 0.003**	**0.718 ± 0.033**
CMLC	0.873 ± 0.021	0.612 ± 0.010
CMSMLC	0.849 ± 0.022	0.686 ± 0.008

C.1.4 Embedding Dimension, $E = 256$

In this scenario, we find that CMLC consistently outperforms all methods when evaluated on Chapman. Similar to findings at lower embedding dimensions, CMSC outperforms all methods on PhysioNet 2020. These claims are supported by the bold test AUC values in Table 10.

Table 10: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.742 ± 0.031	0.591 ± 0.007
CMSC	0.832 ± 0.128	**0.721 ± 0.016**
CMLC	**0.883 ± 0.009**	0.607 ± 0.027
CMSMLC	0.828 ± 0.040	0.652 ± 0.023

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.749 ± 0.032	0.615 ± 0.010
CMSC	0.833 ± 0.130	**0.722 ± 0.017**
CMLC	**0.887 ± 0.008**	0.619 ± 0.026
CMSMLC	0.831 ± 0.042	0.670 ± 0.018

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.752 ± 0.033	0.619 ± 0.010
CMSC	0.833 ± 0.130	**0.723 ± 0.017**
CMLC	**0.889 ± 0.007**	0.626 ± 0.026
CMSMLC	0.831 ± 0.040	0.675 ± 0.018

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.753 ± 0.033	0.621 ± 0.010
CMSC	0.833 ± 0.132	**0.724 ± 0.017**
CMLC	**0.890 ± 0.006**	0.633 ± 0.026
CMSMLC	0.832 ± 0.039	0.677 ± 0.018

C.2 Pre-training and Evaluating using 12 leads

We present Tables 11 - 14 which illustrate the test AUC of an MLR evaluated on Chapman and PhysioNet 2020 after having pre-trained on these two datasets using all 12 leads, respectively, These are presented for a range of embedding dimensions, $E = (32, 64, 128, 256)$, and available labelled training data, $F = (0.25, 0.50, 0.75, 1)$. Overall, we find that pre-training and evaluating with all 12 leads results in a clearly superior self-supervised method, CMSC. We support this claim with the results presented in the subsequent sections. This finding is in contrast to what we observed when pre-training and evaluating on only 4 of the 12 leads. In that scenario, although our proposed methods outperform SimCLR, CMSC does not consistently outperform the other methods.

C.2.1 Embedding Dimension, $E = 32$

We show that CMSC consistently outperforms all other pre-training methods when evaluated on both the Chapman and PhysioNet 2020 dataset. This can be seen by the higher AUC achieved by this method relative to the remaining methods. For instance, when evaluating on the Chapman dataset using only 25% of the labels ($F = 0.25$) during training, CMSC achieves an $\mathrm{AUC} = 0.899$ compared to 0.667 for SimCLR.

Table 11: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.667 ± 0.019	0.585 ± 0.013
CMSC	**0.899 ± 0.003**	**0.744 ± 0.011**
CMLC	0.728 ± 0.021	0.627 ± 0.037
CMSMLC	0.838 ± 0.015	0.644 ± 0.026

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.667 ± 0.021	0.585 ± 0.015
CMSC	**0.898 ± 0.004**	**0.741 ± 0.011**
CMLC	0.729 ± 0.019	0.627 ± 0.038
CMSMLC	0.838 ± 0.018	0.641 ± 0.030

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.667 ± 0.020	0.589 ± 0.014
CMSC	**0.897 ± 0.004**	**0.747 ± 0.012**
CMLC	0.730 ± 0.019	0.635 ± 0.034
CMSMLC	0.843 ± 0.018	0.652 ± 0.025

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.667 ± 0.019	0.589 ± 0.013
CMSC	**0.897 ± 0.004**	**0.745 ± 0.013**
CMLC	0.726 ± 0.019	0.632 ± 0.038
CMSMLC	0.844 ± 0.015	0.649 ± 0.027

C.2.2 Embedding Dimension, $E = 64$

We find that the conclusions arrived at with $E = 32$ are similar to those in this scenario. Namely, CMSC outperforms all remaining methods when evaluating on both Chapman and PhysioNet 2020. Moreover, our other proposed pre-training methods (CMLC and CMSMLC) also outperform the state-of-the-art method, SimCLR. This can be seen by the bold test AUC values in Table 12.

Table 12: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.752 ± 0.045	0.611 ± 0.009
CMSC	**0.904 ± 0.005**	**0.764 ± 0.022**
CMLC	0.734 ± 0.021	0.650 ± 0.039
CMSMLC	0.852 ± 0.024	0.669 ± 0.016

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.753 ± 0.046	0.613 ± 0.011
CMSC	**0.905 ± 0.005**	**0.759 ± 0.024**
CMLC	0.735 ± 0.020	0.650 ± 0.039
CMSMLC	0.851 ± 0.023	0.665 ± 0.016

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.753 ± 0.046	0.617 ± 0.008
CMSC	**0.904 ± 0.005**	**0.770 ± 0.019**
CMLC	0.735 ± 0.020	0.661 ± 0.035
CMSMLC	0.854 ± 0.018	0.675 ± 0.016

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.754 ± 0.045	0.616 ± 0.009
CMSC	**0.905 ± 0.005**	**0.770 ± 0.019**
CMLC	0.735 ± 0.020	0.654 ± 0.040
CMSMLC	0.853 ± 0.017	0.674 ± 0.016

C.2.3 Embedding Dimension, $E = 128$

In this scenario, the same conclusions as those arrived at with smaller embedding dimensions still hold. Although CMSC continues to outperform all other methods, the performance gap between such methods decreases when compared to results obtained at smaller embedding dimensions. For instance, when evaluating on the Chapman dataset at $F = 0.25$ with $E = 128$, CMSC achieves an $\mathrm{AUC} = 0.903$ whereas SimCLR achieves an $\mathrm{AUC} = 0.771$, a performance gap of 13.2%. In contrast, at $E = 64$, the performance gap between these two methods was 15.2%.

Table 13: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.771 ± 0.012	0.605 ± 0.013
CMSC	**0.903 ± 0.002**	**0.7600 ± 0.019**
CMLC	0.779 ± 0.018	0.667 ± 0.030
CMSMLC	0.846 ± 0.024	0.659 ± 0.016

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.773 ± 0.012	0.606 ± 0.013
CMSC	**0.902 ± 0.003**	**0.758 ± 0.019**
CMLC	0.783 ± 0.020	0.665 ± 0.032
CMSMLC	0.850 ± 0.022	0.659 ± 0.016

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.774 ± 0.012	0.611 ± 0.012
CMSC	**0.902 ± 0.003**	**0.763 ± 0.019**
CMLC	0.788 ± 0.018	0.671 ± 0.032
CMSMLC	0.851 ± 0.019	0.669 ± 0.013

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.775 ± 0.012	0.610 ± 0.013
CMSC	**0.902 ± 0.003**	**0.761 ± 0.019**
CMLC	0.787 ± 0.020	0.672 ± 0.028
CMSMLC	0.853 ± 0.017	0.669 ± 0.013

C.2.4 Embedding Dimension, $E = 256$

In this scenario, and similar to findings at lower embedding dimensions, CMSC outperforms all methods on both the Chapman and PhysioNet 2020 datasets. Pre-training and evaluating multi-lead datasets* using all 12 leads. These claims are supported by the bold test AUC values in Table 14.

Table 14: Comparison of self-supervised methods when using networks as feature extractors and performing linear evaluation on downstream datasets. Pre-training and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.769 ± 0.028	0.614 ± 0.007
CMSC	**0.904 ± 0.002**	**0.761 ± 0.011**
CMLC	0.784 ± 0.013	0.672 ± 0.033
CMSMLC	0.852 ± 0.013	0.672 ± 0.013

(b) $F = 0.5$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.770 ± 0.027	0.617 ± 0.008
CMSC	**0.906 ± 0.002**	**0.756 ± 0.010**
CMLC	0.790 ± 0.016	0.668 ± 0.036
CMSMLC	0.852 ± 0.010	0.672 ± 0.012

(c) $F = 0.75$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.770 ± 0.026	0.619 ± 0.008
CMSC	**0.905 ± 0.003**	**0.764 ± 0.011**
CMLC	0.793 ± 0.019	0.680 ± 0.029
CMSMLC	0.854 ± 0.012	0.680 ± 0.012

(d) $F = 1$

Dataset	Chapman*	PhysioNet 2020*
SimCLR	0.771 ± 0.027	0.619 ± 0.008
CMSC	**0.906 ± 0.003**	**0.764 ± 0.010**
CMLC	0.797 ± 0.016	0.677 ± 0.029
CMSMLC	0.858 ± 0.011	0.679 ± 0.011

D Transfer Capabilities of Representations

In this section, we evaluate the utility of self-supervised pre-training in generating a favourable parameter initialization for a downstream task. After pre-training, we transfer the parameters to a downstream task and allow all parameters to be updated. In doing so, we are evaluating the benefit brought about by the inductive bias of self-supervised pre-training.

We perform these experiments under two scenarios. The first involves pre-training, fine-tuning, and evaluating using 4 leads (II, V2, aVL, aVR) (see Sec. D.1). The second involves pre-training, fine-tuning, and evaluating using all 12 leads (see Sec. D.2). We chose these two scenarios for several reasons. Firstly, they will help determine whether our findings generalize to domains where a different number of leads is available. For example, expensive hospital equipment may record all 12 leads of an ECG, whereas low-cost wearable sensors may only collect data from a subset of leads. Secondly, we wanted to evaluate whether or not contrastive-learning with more views (leads) would improve generalization performance on the downstream task. Previous studies in computer vision have shown this to be the case.

D.1 Pre-training, Fine-Tuning, and Evaluating using 4 Leads

We present Tables 15 - 18 which illustrate the test AUC on downstream datasets after having pre-trained on Chapman or PhysioNet 2020. These are shown for a range of embedding dimensions, $E = (32, 64, 128, 256)$, and available labelled training data, $F = (0.25, 0.50, 0.75, 1.00)$.

D.1.1 Embedding Dimension, $E = 32$

In this section, we show that in 18/24 (75%) of all experiments conducted, our family of contrastive learning methods outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 15. The majority of these positive results can be attributed to CMSC. Such a finding illustrates the robustness of our methods to the pre-training and downstream dataset used for evaluation, especially given the diversity of the tasks at hand.

Table 15: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.631 ± 0.006	0.738 ± 0.014	0.766 ± 0.005	0.631 ± 0.006	0.738 ± 0.014	0.898 ± 0.002
SimCLR	0.649 ± 0.012	0.731 ± 0.017	0.790± 0.008	0.642 ± 0.020	0.738 ± 0.009	0.907 ± 0.013
CMSC	0.661 ± 0.018	**0.770 ± 0.012**	**0.801 ± 0.013**	**0.658 ± 0.018**	0.748 ± 0.027	**0.908 ± 0.011**
CMLC	0.652 ± 0.014	0.767 ± 0.012	0.768 ± 0.004	0.635 ± 0.017	**0.753 ± 0.013**	0.906 ± 0.009
CMSMLC	**0.669 ± 0.020**	0.758 ± 0.008	0.761 ± 0.015	0.652 ± 0.013	0.733 ± 0.004	0.900 ± 0.009

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.669 ± 0.007	0.782 ± 0.011	0.811 ± 0.011	0.669 ± 0.007	0.782 ± 0.011	0.907 ± 0.011
SimCLR	0.691 ± 0.008	0.748 ± 0.018	**0.829 ± 0.003**	0.679 ± 0.012	0.767 ± 0.012	**0.933 ± 0.010**
CMSC	0.687 ± 0.018	**0.771 ± 0.030**	0.822 ± 0.011	**0.689 ± 0.025**	**0.769 ± 0.010**	0.926 ± 0.010
CMLC	0.680 ± 0.003	0.772 ± 0.007	0.812 ± 0.013	0.677 ± 0.013	0.764 ± 0.027	0.918 ± 0.008
CMSMLC	**0.708 ± 0.017**	0.769 ± 0.015	0.799 ± 0.011	0.684 ± 0.011	0.761 ± 0.022	0.923 ± 0.012

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.682 ± 0.016	0.764 ± 0.011	0.824 ± 0.013	0.682 ± 0.016	0.764 ± 0.011	0.925 ± 0.009
SimCLR	0.699 ± 0.010	0.782 ± 0.015	**0.839 ± 0.003**	0.691 ± 0.009	**0.795 ± 0.017**	0.938 ± 0.012
CMSC	0.712 ± 0.011	0.760 ± 0.032	0.835 ± 0.006	**0.704 ± 0.024**	0.780 ± 0.015	**0.941 ± 0.006**
CMLC	0.682 ± 0.011	0.769 ± 0.020	0.826 ± 0.014	0.665 ± 0.016	0.764 ± 0.020	0.930 ± 0.013
CMSMLC	**0.715 ± 0.009**	**0.789 ± 0.014**	0.820 ± 0.004	0.703 ± 0.010	0.778 ± 0.019	0.936 ± 0.007

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.700 ± 0.019	0.771 ± 0.018	0.832 ± 0.006	0.700 ± 0.019	0.771 ± 0.018	0.937 ± 0.005
SimCLR	0.715 ± 0.005	**0.804 ± 0.020**	**0.844 ± 0.001**	0.704 ± 0.009	0.785 ± 0.025	0.938 ± 0.011
CMSC	**0.723 ± 0.004**	0.803 ± 0.033	0.841 ± 0.007	**0.724 ± 0.015**	**0.795 ± 0.009**	**0.945 ± 0.004**
CMLC	0.699 ± 0.025	0.778 ± 0.015	0.837 ± 0.006	0.707 ± 0.014	0.780 ± 0.023	0.933 ± 0.014
CMSMLC	0.719 ± 0.016	0.798 ± 0.018	0.830 ± 0.006	0.715 ± 0.014	0.775 ± 0.009	0.940 ± 0.006

D.1.2 Embedding Dimension, $E = 64$

In this section, we show that in 20/24 (83%) of all experiments conducted, our family of contrastive learning methods outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 16.

Table 16: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.632 ± 0.018	0.746 ± 0.009	0.775 ± 0.010	0.632 ± 0.018	0.746 ± 0.009	0.895 ± 0.001
SimCLR	0.652 ± 0.010	0.744 ± 0.013	0.784 ± 0.022	0.641 ± 0.006	0.739 ± 0.019	**0.911 ± 0.007**
CMSC	0.663 ± 0.019	**0.765 ± 0.023**	**0.794 ± 0.022**	0.659 ± 0.032	0.755 ± 0.018	0.910 ± 0.007
CMLC	0.650 ± 0.007	0.753 ± 0.012	0.786 ± 0.008	0.644 ± 0.014	**0.762 ± 0.009**	0.904 ± 0.007
CMSMLC	**0.675 ± 0.010**	0.755 ± 0.009	0.767 ± 0.008	**0.660 ± 0.012**	0.743 ± 0.016	0.901 ± 0.003

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.685 ± 0.004	0.768 ± 0.010	0.817 ± 0.009	0.685 ± 0.004	0.768 ± 0.010	0.906 ± 0.003
SimCLR	0.676 ± 0.019	0.778 ± 0.008	0.822 ± 0.011	0.678 ± 0.009	0.771 ± 0.018	0.927 ± 0.009
CMSC	0.695 ± 0.011	**0.786 ± 0.017**	0.816 ± 0.016	**0.701 ± 0.023**	0.772 ± 0.009	0.928 ± 0.004
CMLC	0.679 ± 0.016	**0.775 ± 0.010**	**0.824 ± 0.004**	0.677 ± 0.021	**0.775 ± 0.010**	0.918 ± 0.014
CMSMLC	**0.717 ± 0.005**	0.773 ± 0.011	0.808 ± 0.009	0.699 ± 0.011	0.769 ± 0.009	**0.934 ± 0.004**

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.680 ± 0.015	0.7600 ± 0.011	0.830 ± 0.007	0.680 ± 0.015	0.7600 ± 0.011	0.916 ± 0.011
SimCLR	0.698 ± 0.008	0.790 ± 0.010	0.832 ± 0.007	0.689 ± 0.016	**0.783 ± 0.015**	0.934 ± 0.006
CMSC	0.708 ± 0.006	0.790 ± 0.027	0.834 ± 0.004	**0.715 ± 0.008**	0.779 ± 0.013	**0.940 ± 0.007**
CMLC	0.688 ± 0.016	0.777 ± 0.017	**0.837 ± 0.003**	0.678 ± 0.019	0.777 ± 0.011	0.926 ± 0.016
CMSMLC	**0.720 ± 0.004**	**0.795 ± 0.008**	0.829 ± 0.009	0.704 ± 0.007	0.775 ± 0.013	0.932 ± 0.005

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.692 ± 0.023	0.778 ± 0.017	0.840 ± 0.004	0.692 ± 0.023	0.778 ± 0.017	0.932 ± 0.008
SimCLR	0.718 ± 0.010	0.797 ± 0.014	0.837 ± 0.009	0.706 ± 0.008	0.789 ± 0.023	**0.944 ± 0.004**
CMSC	0.708 ± 0.022	0.800 ± 0.020	**0.842 ± 0.003**	**0.726 ± 0.007**	**0.797 ± 0.009**	0.941 ± 0.005
CMLC	0.691 ± 0.007	0.780 ± 0.013	0.841 ± 0.003	0.696 ± 0.022	0.786 ± 0.016	0.931 ± 0.018
CMSMLC	**0.732 ± 0.003**	**0.810 ± 0.012**	0.837 ± 0.010	0.722 ± 0.010	0.792 ± 0.006	0.940 ± 0.007

D.1.3 Embedding Dimension, $E = 128$

In this section, we show that in 21/24 (88%) of all experiments conducted, our family of contrastive learning methods outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 17.

Table 17: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.625 ± 0.015	0.746 ± 0.006	0.764 ± 0.016	0.625 ± 0.015	0.746 ± 0.006	0.894 ± 0.002
SimCLR	0.634 ± 0.014	0.738 ± 0.006	0.777 ± 0.015	0.631 ± 0.022	0.727 ± 0.014	0.903 ± 0.007
CMSC	**0.691 ± 0.015**	**0.768 ± 0.005**	**0.813 ± 0.007**	**0.671 ± 0.018**	**0.756 ± 0.009**	**0.911 ± 0.016**
CMLC	0.639 ± 0.010	0.745 ± 0.012	0.770 ± 0.006	0.641 ± 0.014	0.746 ± 0.014	0.897 ± 0.003
CMSMLC	0.671 ± 0.016	0.755 ± 0.011	0.781 ± 0.012	0.668 ± 0.011	0.751 ± 0.007	0.903 ± 0.009

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.678 ± 0.011	0.763 ± 0.005	0.803 ± 0.008	0.678 ± 0.011	0.763 ± 0.005	0.907 ± 0.006
SimCLR	0.676 ± 0.011	0.772 ± 0.010	0.823 ± 0.011	0.658 ± 0.027	0.762 ± 0.009	0.923 ± 0.010
CMSC	0.695 ± 0.024	0.773 ± 0.013	**0.830 ± 0.002**	**0.714 ± 0.014**	0.760 ± 0.013	**0.932 ± 0.008**
CMLC	0.665 ± 0.016	0.767 ± 0.013	0.810 ± 0.011	0.675 ± 0.013	0.762 ± 0.007	0.910 ± 0.012
CMSMLC	**0.717 ± 0.006**	**0.774 ± 0.004**	0.814 ± 0.009	0.698 ± 0.011	**0.774 ± 0.012**	0.930 ± 0.012

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.675 ± 0.020	0.775 ± 0.005	0.831 ± 0.011	0.675 ± 0.020	0.775 ± 0.005	0.937 ± 0.008
SimCLR	0.694 ± 0.019	0.776 ± 0.013	0.834 ± 0.009	0.686 ± 0.019	**0.785 ± 0.011**	0.931 ± 0.013
CMSC	0.700 ± 0.012	**0.801 ± 0.013**	**0.840 ± 0.004**	0.707 ± 0.015	0.777 ± 0.016	**0.942 ± 0.012**
CMLC	0.670 ± 0.019	0.771 ± 0.010	0.831 ± 0.004	0.682 ± 0.005	0.772 ± 0.009	0.917 ± 0.011
CMSMLC	**0.719 ± 0.011**	0.792 ± 0.014	0.837 ± 0.008	**0.711 ± 0.011**	0.777 ± 0.017	0.938 ± 0.010

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.702 ± 0.016	0.773 ± 0.010	0.843 ± 0.002	0.702 ± 0.016	0.773 ± 0.010	0.930 ± 0.013
SimCLR	0.705 ± 0.008	**0.810 ± 0.016**	0.844 ± 0.005	0.700 ± 0.012	**0.795 ± 0.021**	0.941 ± 0.006
CMSC	0.715 ± 0.018	0.804 ± 0.018	**0.846 ± 0.002**	**0.725 ± 0.020**	0.779 ± 0.024	0.942 ± 0.009
CMLC	0.698 ± 0.007	0.781 ± 0.014	0.836 ± 0.003	0.681 ± 0.005	0.785 ± 0.011	0.933 ± 0.014
CMSMLC	**0.732 ± 0.003**	0.793 ± 0.012	0.844 ± 0.005	0.716 ± 0.010	0.778 ± 0.025	**0.945 ± 0.005**

D.1.4 Embedding Dimension, $E = 256$

In this section, we show that in 16/24 (66%) of all experiments conducted, our family of contrastive learning methods outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 18.

Table 18: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using 4 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.630 ± 0.014	0.737 ± 0.008	0.765 ± 0.004	0.630 ± 0.014	0.737 ± 0.008	0.896 ± 0.002
SimCLR	0.647 ± 0.014	0.727 ± 0.007	**0.791 ± 0.014**	0.636 ± 0.009	0.736 ± 0.008	0.902 ± 0.006
CMSC	0.656 ± 0.031	**0.756 ± 0.011**	0.789 ± 0.019	**0.682 ± 0.024**	0.750 ± 0.014	**0.905 ± 0.009**
CMLC	0.649 ± 0.012	0.743 ± 0.005	0.784 ± 0.009	0.645 ± 0.017	0.741 ± 0.008	0.898 ± 0.004
CMSMLC	**0.686 ± 0.008**	0.752 ± 0.010	0.768 ± 0.017	0.652 ± 0.023	**0.758 ± 0.014**	0.896 ± 0.002

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.659 ± 0.012	0.758 ± 0.021	0.817 ± 0.008	0.659 ± 0.012	0.758 ± 0.021	0.901 ± 0.003
SimCLR	0.667 ± 0.019	0.758 ± 0.002	0.825 ± 0.014	0.659 ± 0.010	**0.769 ± 0.017**	**0.924 ± 0.012**
CMSC	0.667 ± 0.030	0.765 ± 0.003	0.819 ± 0.002	**0.709 ± 0.028**	0.762 ± 0.015	0.914 ± 0.011
CMLC	0.679 ± 0.014	0.768 ± 0.006	**0.826 ± 0.005**	0.669 ± 0.027	0.768 ± 0.012	0.906 ± 0.007
CMSMLC	**0.702 ± 0.017**	**0.776 ± 0.011**	0.812 ± 0.014	0.694 ± 0.011	0.762 ± 0.009	0.917 ± 0.011

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.680 ± 0.018	0.764 ± 0.006	0.834 ± 0.004	0.680± 0.018	0.764 ± 0.006	0.916 ± 0.015
SimCLR	0.677 ± 0.016	**0.790 ± 0.015**	0.834 ± 0.011	0.684 ± 0.008	**0.787 ± 0.015**	0.933 ± 0.013
CMSC	0.698 ± 0.015	0.784 ± 0.015	0.827 ± 0.014	**0.717 ± 0.010**	0.780 ± 0.018	**0.935 ± 0.007**
CMLC	0.677 ± 0.023	0.773 ± 0.006	**0.841 ± 0.001**	0.681 ± 0.012	0.779 ± 0.012	0.917 ± 0.012
CMSMLC	**0.715 ± 0.008**	0.785 ± 0.004	0.827 ± 0.015	0.697 ± 0.016	0.784 ± 0.010	0.930 ± 0.004

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.696 ± 0.015	0.763 ± 0.012	0.842 ± 0.005	0.696 ± 0.015	0.763 ± 0.012	0.918 ± 0.015
SimCLR	0.711 ± 0.008	**0.798 ± 0.014**	0.841 ± 0.006	0.703 ± 0.007	**0.806 ± 0.012**	0.943 ± 0.004
CMSC	0.704 ± 0.023	0.794 ± 0.018	0.840 ± 0.007	**0.718 ± 0.012**	0.792 ± 0.016	**0.944 ± 0.007**
CMLC	0.705 ± 0.009	0.781 ± 0.005	**0.844 ± 0.002**	0.690 ± 0.020	0.779 ± 0.007	0.926 ± 0.015
CMSMLC	**0.731 ± 0.007**	0.789 ± 0.020	0.839 ± 0.007	0.709 ± 0.013	0.791 ± 0.007	0.943 ± 0.004

D.2 Pre-training, Fine-tuning, and Evaluating using 12 Leads

We present Tables 19 - 22 which illustrate the test AUC on downstream datasets after having pre-trained on Chapman or PhysioNet 2020 using all 12 leads. These are shown for a range of embedding dimensions, $E = (32, 64, 128, 256)$, and available labelled training data, $F = (0.25, 0.50, 0.75, 1.00)$. Overall, we find that encouraging the representations of a large and diverse set of leads to be similar to one another might be detrimental. This is shown in the subsequent sections by the consistently poorer performance ($\downarrow$ AUC) of CMLC and CMSMLC relative to CMSC where the latter method does not enforce the aforementioned similarity.

D.2.1 Embedding Dimension, $E = 32$

In this section, we show that in 22/24 (92%) of all experiments conducted, CMSC outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 19. Such a finding illustrates the robustness of our methods to the pre-training and downstream dataset used for evaluation, especially given the diversity of the tasks at hand.

The performance gap between CMSC and SimCLR widens as the fraction of available labelled training data decreases. For instance, when evaluating on the Cardiology dataset, as $F = 1 \rightarrow 0.25$, CMSC's $\text{AUC} = 0.723 \rightarrow 0.689$ whereas SimCLR's $\text{AUC} = 0.694 \rightarrow 0.636$. Therefore, the performance gap widens by almost a factor of 2 from 2.9% to 5.3%. This suggests that CMSC is better equipped to deal with downstream tasks that lack a sufficient amount of labelled data.

Table 19: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.631 ± 0.006	0.738 ± 0.014	0.823 ± 0.007	0.631 ± 0.006	0.738 ± 0.014	0.907 ± 0.006
SimCLR	0.636 ± 0.019	0.724 ± 0.016	0.826 ± 0.011	0.616 ± 0.011	0.727 ± 0.020	0.921 ± 0.011
CMSC	**0.689 ± 0.017**	**0.782 ± 0.005**	**0.833 ± 0.002**	**0.681 ± 0.017**	**0.769 ± 0.015**	**0.936 ± 0.011**
CMLC	0.639 ± 0.023	0.744 ± 0.018	0.827 ± 0.003	0.630 ± 0.022	0.744 ± 0.022	0.912 ± 0.007
CMSMLC	0.644 ± 0.026	0.740 ± 0.019	0.818 ± 0.015	0.647 ± 0.022	0.745 ± 0.015	0.920 ± 0.011

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.669 ± 0.007	0.782 ± 0.011	0.814 ± 0.009	0.669 ± 0.007	0.782 ± 0.011	0.938 ± 0.009
SimCLR	0.659 ± 0.011	0.764 ± 0.003	0.820 ± 0.032	0.669 ± 0.023	0.766 ± 0.015	0.936 ± 0.014
CMSC	0.686 ± 0.024	**0.800 ± 0.013**	**0.836 ± 0.004**	**0.719 ± 0.014**	**0.778 ± 0.019**	**0.951 ± 0.003**
CMLC	0.674 ± 0.012	0.773 ± 0.018	0.831 ± 0.002	0.667 ± 0.011	0.758 ± 0.017	0.933 ± 0.008
CMSMLC	**0.691 ± 0.007**	0.759 ± 0.020	0.831 ± 0.009	0.684 ± 0.028	0.763 ± 0.024	0.942 ± 0.005

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.682 ± 0.016	0.764 ± 0.011	0.845 ± 0.001	0.682 ± 0.016	0.764 ± 0.011	0.937 ± 0.016
SimCLR	0.690 ± 0.023	0.786 ± 0.023	0.840 ± 0.006	0.668 ± 0.013	0.782 ± 0.007	0.945 ± 0.009
CMSC	0.702 ± 0.013	**0.809 ± 0.009**	**0.847 ± 0.001**	**0.709 ± 0.010**	**0.806 ± 0.005**	**0.952 ± 0.010**
CMLC	0.684 ± 0.027	0.774 ± 0.019	0.841 ± 0.015	0.680 ± 0.021	0.783 ± 0.018	0.933 ± 0.014
CMSMLC	**0.719 ± 0.007**	0.757 ± 0.025	0.843 ± 0.005	0.708 ± 0.011	0.787 ± 0.015	0.944 ± 0.006

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.700 ± 0.019	0.771 ± 0.018	0.825 ± 0.016	0.700 ± 0.019	0.771 ± 0.018	0.945 ± 0.003
SimCLR	0.694 ± 0.010	0.790 ± 0.022	0.839 ± 0.008	0.691 ± 0.009	0.790 ± 0.020	0.942 ± 0.014
CMSC	**0.723 ± 0.011**	**0.821 ± 0.013**	**0.845 ± 0.003**	**0.725 ± 0.017**	**0.798 ± 0.008**	**0.954 ± 0.007**
CMLC	0.702 ± 0.011	0.762 ± 0.014	0.844 ± 0.003	0.708 ± 0.026	0.777 ± 0.019	0.948 ± 0.005
CMSMLC	0.722 ± 0.007	0.782 ± 0.013	0.845 ± 0.005	0.710 ± 0.020	0.768 ± 0.033	0.946 ± 0.005

D.2.2 Embedding Dimension, $E = 64$

In this section, we show that in 21/24 (88%) of all experiments conducted, CMSC outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 20.

Table 20: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.632 ± 0.018	0.746 ± 0.009	0.822 ± 0.011	0.632 ± 0.018	0.746 ± 0.009	0.901 ± 0.004
SimCLR	0.632 ± 0.021	0.736 ± 0.019	0.833 ± 0.008	0.626 ± 0.008	0.734 ± 0.018	0.925 ± 0.013
CMSC	**0.681 ± 0.024**	**0.798 ± 0.008**	**0.834 ± 0.006**	**0.658 ± 0.026**	**0.779 ± 0.012**	**0.942 ± 0.011**
CMLC	0.626 ± 0.025	0.735 ± 0.011	0.825 ± 0.004	0.627 ± 0.016	0.739 ± 0.014	0.910 ± 0.007
CMSMLC	0.659 ± 0.024	0.738 ± 0.013	0.820 ± 0.016	0.647 ± 0.023	0.743 ± 0.012	0.912 ± 0.009

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.685 ± 0.004	0.768 ± 0.010	0.831 ± 0.007	0.685 ± 0.004	0.768 ± 0.01	0.931 ± 0.016
SimCLR	0.672 ± 0.023	0.762 ± 0.021	0.833 ± 0.011	0.681 ± 0.011	0.767 ± 0.012	0.943 ± 0.006
CMSC	**0.708 ± 0.010**	**0.804 ± 0.011**	**0.834 ± 0.010**	**0.709 ± 0.013**	**0.792 ± 0.015**	**0.954 ± 0.005**
CMLC	0.680 ± 0.017	0.763 ± 0.010	0.832 ± 0.005	0.694 ± 0.019	0.748 ± 0.023	0.933 ± 0.009
CMSMLC	0.706 ± 0.007	0.759 ± 0.014	0.815 ± 0.025	0.699 ± 0.023	0.753 ± 0.017	0.940 ± 0.008

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.68 ± 0.015	0.76 ± 0.011	0.841 ± 0.008	0.680 ± 0.015	0.760 ± 0.011	0.937 ± 0.009
SimCLR	0.695 ± 0.023	0.779 ± 0.012	0.844 ± 0.007	0.674 ± 0.017	0.775 ± 0.011	0.948 ± 0.009
CMSC	0.709 ± 0.014	**0.809 ± 0.014**	0.844 ± 0.007	**0.714 ± 0.017**	**0.802 ± 0.012**	**0.953 ± 0.006**
CMLC	0.690± 0.007	0.778 ± 0.010	**0.844 ± 0.002**	0.704 ± 0.021	0.768 ± 0.018	0.946 ± 0.003
CMSMLC	**0.711 ± 0.011**	0.763 ± 0.016	0.838 ± 0.006	0.689 ± 0.022	0.762 ± 0.019	0.946 ± 0.008

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.692 ± 0.023	0.778 ± 0.017	0.846 ± 0.003	0.692 ± 0.023	0.778 ± 0.017	0.946 ± 0.005
SimCLR	0.715 ± 0.011	0.808 ± 0.009	0.842 ± 0.007	0.703 ± 0.006	0.797 ± 0.018	0.952 ± 0.008
CMSC	**0.736 ± 0.016**	**0.810 ± 0.005**	0.843 ± 0.005	**0.731 ± 0.010**	**0.810 ± 0.015**	**0.958 ± 0.007**
CMLC	0.706 ± 0.012	0.777 ± 0.017	**0.846 ± 0.002**	0.709 ± 0.012	0.779 ± 0.018	0.947 ± 0.005
CMSMLC	0.722 ± 0.008	0.780 ± 0.015	0.842 ± 0.008	0.701 ± 0.023	0.779 ± 0.015	0.943 ± 0.009

D.2.3 Embedding Dimension, $E = 128$

In this section, we show that in 24/24 (100%) of all experiments conducted, CMSC outperforms the the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 21.

Table 21: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.625 ± 0.015	0.746 ± 0.006	0.819 ± 0.008	0.625 ± 0.015	0.746 ± 0.006	0.909 ± 0.006
SimCLR	0.630 ± 0.011	0.735 ± 0.012	0.833 ± 0.008	0.624 ± 0.007	0.729 ± 0.018	0.918 ± 0.015
CMSC	$\mathbf{0.678 \pm 0.010}$	$\mathbf{0.790 \pm 0.012}$	$\mathbf{0.833 \pm 0.008}$	$\mathbf{0.680 \pm 0.011}$	$\mathbf{0.777 \pm 0.027}$	$\mathbf{0.940 \pm 0.007}$
CMLC	0.639 ± 0.012	0.740 ± 0.007	0.831 ± 0.003	0.639 ± 0.019	0.743 ± 0.016	0.913 ± 0.012
CMSMLC	0.661 ± 0.029	0.748 ± 0.005	0.813 ± 0.024	0.646 ± 0.023	0.736 ± 0.007	0.918 ± 0.012

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.678 ± 0.011	0.763 ± 0.005	0.832 ± 0.003	0.678 ± 0.011	0.763 ± 0.005	0.931 ± 0.014
SimCLR	0.667 ± 0.021	0.768 ± 0.012	0.835 ± 0.010	0.659 ± 0.012	0.754 ± 0.024	0.939 ± 0.007
CMSC	$\mathbf{0.716 \pm 0.010}$	$\mathbf{0.802 \pm 0.007}$	$\mathbf{0.840 \pm 0.003}$	$\mathbf{0.718 \pm 0.005}$	$\mathbf{0.791 \pm 0.025}$	$\mathbf{0.944 \pm 0.008}$
CMLC	0.690 ± 0.012	0.763 ± 0.009	0.840 ± 0.003	0.663 ± 0.040	0.752 ± 0.016	0.927 ± 0.013
CMSMLC	0.699 ± 0.013	0.751 ± 0.013	0.815 ± 0.014	0.695 ± 0.020	0.748 ± 0.013	0.931 ± 0.011

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.675 ± 0.020	0.775 ± 0.005	0.844 ± 0.006	0.675 ± 0.020	0.775 ± 0.005	0.945 ± 0.004
SimCLR	0.682 ± 0.023	0.775 ± 0.009	0.843 ± 0.007	0.681 ± 0.020	0.764 ± 0.019	0.946 ± 0.010
CMSC	$\mathbf{0.719 \pm 0.008}$	$\mathbf{0.813 \pm 0.006}$	$\mathbf{0.847 \pm 0.002}$	$\mathbf{0.711 \pm 0.004}$	$\mathbf{0.810 \pm 0.020}$	$\mathbf{0.955 \pm 0.005}$
CMLC	0.684 ± 0.008	0.777 ± 0.021	0.846 ± 0.001	0.700 ± 0.016	0.755 ± 0.016	0.942 ± 0.005
CMSMLC	0.711 ± 0.011	0.782 ± 0.006	0.839 ± 0.007	0.694 ± 0.028	0.769 ± 0.014	0.941 ± 0.007

(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.702 ± 0.016	0.773 ± 0.01	0.842 ± 0.008	0.702 ± 0.016	0.773 ± 0.01	0.942 ± 0.006
SimCLR	0.703 ± 0.020	0.801 ± 0.014	0.845 ± 0.009	0.703 ± 0.014	0.784 ± 0.009	0.948 ± 0.008
CMSC	$\mathbf{0.731 \pm 0.022}$	$\mathbf{0.819 \pm 0.004}$	$\mathbf{0.847 \pm 0.003}$	$\mathbf{0.718 \pm 0.012}$	$\mathbf{0.809 \pm 0.021}$	$\mathbf{0.959 \pm 0.004}$
CMLC	0.705 ± 0.010	0.777 ± 0.011	0.845 ± 0.002	0.713 ± 0.023	0.789 ± 0.012	0.946 ± 0.005
CMSMLC	0.719 ± 0.005	0.764 ± 0.010	0.837 ± 0.007	0.711 ± 0.013	0.779 ± 0.013	0.947 ± 0.003

D.2.4 Embedding Dimension, $E = 256$

In this section, we show that in 22/24 (92%) of all experiments conducted, CMSC outperforms the state-of-the-art method, SimCLR. This can be seen by the bold test AUC results in Table 22.

Table 22: Comparison of self-supervised methods when used as parameter initializations before fine-tuning on downstream datasets. Pre-training, fine-tuning, and evaluating multi-lead datasets* using all 12 leads. Mean and standard deviation are shown across 5 seeds.

(a) $F = 0.25$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.630 ± 0.014	0.737 ± 0.008	0.809 ± 0.023	0.630 ± 0.014	0.737 ± 0.008	0.903 ± 0.005
SimCLR	0.620 ± 0.028	0.729 ± 0.013	0.830 ± 0.007	0.621 ± 0.016	0.726 ± 0.008	0.933 ± 0.007
CMSC	**0.692 ± 0.007**	**0.792 ± 0.014**	**0.832 ± 0.009**	**0.689 ± 0.013**	**0.782 ± 0.010**	**0.940 ± 0.010**
CMLC	0.618 ± 0.004	0.733 ± 0.006	0.831 ± 0.009	0.648 ± 0.018	0.743 ± 0.010	0.912 ± 0.006
CMSMLC	0.666 ± 0.012	0.741 ± 0.010	0.820 ± 0.013	0.666 ± 0.008	0.736 ± 0.012	0.922 ± 0.011

(b) $F = 0.5$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.659 ± 0.012	0.758 ± 0.021	0.831 ± 0.011	0.659 ± 0.012	0.758 ± 0.021	0.929 ± 0.010
SimCLR	0.670 ± 0.021	0.764 ± 0.008	0.830 ± 0.011	0.663 ± 0.007	0.762 ± 0.009	0.942 ± 0.005
CMSC	**0.706 ± 0.024**	**0.809 ± 0.004**	0.835 ± 0.009	**0.714 ± 0.006**	**0.798 ± 0.009**	**0.953 ± 0.007**
CMLC	0.668 ± 0.006	0.762 ± 0.005	**0.837 ± 0.007**	0.700 ± 0.013	0.768 ± 0.011	0.935 ± 0.010
CMSMLC	0.704 ± 0.012	0.763 ± 0.009	0.829 ± 0.009	0.713 ± 0.006	0.748 ± 0.011	0.940 ± 0.003

(c) $F = 0.75$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.680 ± 0.018	0.764 ± 0.006	0.844 ± 0.004	0.68 ± 0.018	0.764 ± 0.006	0.936 ± 0.014
SimCLR	0.675 ± 0.016	0.782 ± 0.014	0.842 ± 0.007	0.678 ± 0.007	0.786 ± 0.010	0.953 ± 0.002
CMSC	**0.714 ± 0.013**	**0.816 ± 0.003**	**0.843 ± 0.006**	**0.722 ± 0.015**	**0.805 ± 0.011**	**0.958 ± 0.003**
CMLC	0.678 ± 0.011	0.777 ± 0.009	0.841 ± 0.006	0.705 ± 0.013	0.775 ± 0.013	0.936 ± 0.008
CMSMLC	0.701 ± 0.014	0.775 ± 0.009	0.842 ± 0.007	0.705 ± 0.004	0.763 ± 0.009	0.946 ± 0.005

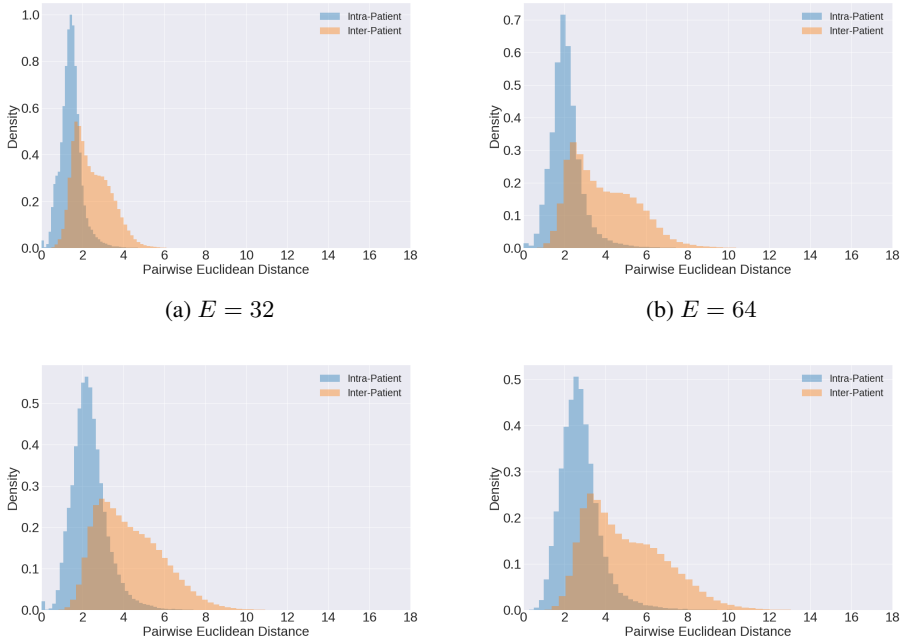
(d) $F = 1$

Pretraining Dataset	Chapman*			PhysioNet 2020*		
Downstream Dataset	Cardiology	PhysioNet 2017	PhysioNet 2020*	Cardiology	PhysioNet 2017	Chapman*
Random Init	0.696 ± 0.015	0.763 ± 0.012	0.839 ± 0.009	0.696 ± 0.015	0.763 ± 0.012	0.943 ± 0.002
SimCLR	0.708 ± 0.019	0.789 ± 0.006	**0.844 ± 0.007**	0.707 ± 0.009	0.792 ± 0.008	0.951 ± 0.005
CMSC	**0.735 ± 0.006**	**0.822 ± 0.004**	0.843 ± 0.006	**0.729 ± 0.010**	**0.807 ± 0.012**	**0.957 ± 0.004**
CMLC	0.705 ± 0.006	0.795 ± 0.014	0.843 ± 0.007	0.719 ± 0.003	0.793 ± 0.016	0.942 ± 0.006
CMSMLC	0.722 ± 0.008	0.778 ± 0.013	0.842 ± 0.008	0.722 ± 0.003	0.767 ± 0.013	0.946 ± 0.003

E Intra and Inter-Patient Representation Distances

A key component of our contrastive approach is the redefinition of the positive pair to refer to instances from the same patient. In order to validate whether our training procedure truly satisfies this goal, we calculate the pairwise distance between representations of instances in two scenarios. The first scenario consists solely of instances from the same patient (Intra-Patient). The second scenario calculates the distance between representations of instances belonging to different patients (Inter-Patient). In Fig. 6, we illustrate the distribution of these distances for instances in the validation set of PhysioNet 2020.

We show that, when using a low embedding dimension ($E = 32$), the intra-patient distances are the lowest with a mean of around 1. As $E = 32 \rightarrow 256$, the distributions begin to shift to higher values. Such high pairwise distances imply that maintaining similar representations at higher dimensions is more difficult. Moreover, we clearly see two distinct distributions belonging to intra-patient and inter-patient distances. This suggests that the training procedure worked as expected, leading to representations that are more similar within patients than across patients.



(a) $E = 32$ (b) $E = 64$

(c) $E = 128$ (d) $E = 256$

Figure 6: Distribution of pairwise Euclidean distance between representations belonging to the same patient (Intra-Patient) and those belonging to different patients (Inter-Patient). Representations were of instances present in the validation set of PhysioNet 2020. Self-supervision was performed with CMSC on PhysioNet 2020 using 4 leads.