Self-Supervised Contrastive Learning for Electrocardiograms to Detect Left Ventricular Systolic Dysfunction

Mitsuhiko Nakamoto
The University of Tokyo
mitsuhiko0820@g.ecc.u-tokyo.ac.jp

Satoshi Kodera
The University of Tokyo
koderas-int@h.u-tokyo.ac.jp

Hirotoshi Takeuchi
The University of Tokyo
takeuchi-hirotoshi742@g.ecc.u-tokyo.ac.jp

Shinnosuke Sawano
The University of Tokyo
sawanos-int@h.u-tokyo.ac.jp

Susumu Katsushika
The University of Tokyo
katsushikas-int@h.u-tokyo.ac.jp

Issei Komuro
The University of Tokyo
komuro-3im@h.u-tokyo.ac.jp

Abstract

Self-supervised learning has been demonstrated to be a powerful way to use unlabeled data in computer vision tasks. In this study, we propose a self-supervised pretraining approach to improve the performance of deep learning models that detect left ventricular systolic dysfunction from 12-lead electrocardiography data. We first pretrain an encoder that can extract rich features from unlabeled electrocardiography data using self-supervised contrastive learning, and then fine-tune the model on the downstream dataset using the pretrained encoder. In experiments, our proposed approach achieved higher performance than the supervised baseline method, using only 28% of the labels used by the baseline method.

1 Introduction

Self-supervised learning has been demonstrated to be a powerful way to use unlabeled data in computer vision tasks [1, 2, 3]. In particular, contrastive learning approaches, such as MoCo [4, 5] and SimCLR [6], have achieved high performance on ImageNet [7] using only a small number of labels. By using these methods, we can first pretrain an encoder that can extract rich features from unlabeled data, and then fine-tune the encoder on a small labeled downstream dataset.

Self-supervised learning is an ideal method for medical fields because collecting high-quality labels for medical data is usually extremely difficult. Effective use of self-supervised learning may achieve high diagnostic accuracy even with small labeled datasets. In this study, we aimed to apply self-supervised contrastive learning to the analysis of electrocardiograms (ECGs), which are among the most basic types of medical images.

The deep learning approach to analyzing ECGs has been widely studied in recent years, enabling highly accurate detection of cardiac diseases [8, 9, 10, 11], which were previously difficult to diagnose from ECGs. However, these approaches require large amounts of labeled electrocardiography data, which are difficult to collect at a single institution. To solve this problem, several methods that apply self-supervised pretraining to ECGs have been proposed [12, 13, 14, 15]. However, no study has ever applied self-supervised pretraining to improve a model for diagnosing left ventricular (LV) systolic

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

dysfunction. LV systolic dysfunction is a common disease that may lead to an increased risk of sudden death, therefore, early detection is important. In clinical practice, LV systolic dysfunction is currently diagnosed using echocardiography and is difficult to diagnose from ECGs.

In this paper, we show that self-supervised contrastive learning can be used to improve the performance of deep learning models that detect LV systolic dysfunction from 12-lead electrocardiography data. We propose several data transformation techniques and an architecture based on a two-dimensional (2D) convolutional neural network (CNN) as an encoder for pretraining using a contrastive learning algorithm. The pretrained encoder is then used to train on a downstream dataset to detect LV systolic dysfunction. Experimental results show that our proposed approach achieved higher performance than the supervised baseline method, using only 28% of the labels used by the baseline method. Finally, we also use Grad-CAM [16] visualization to discover which part of an ECG is the focus of the pretrained encoder.

2 Method

Dataset: We used a dataset of 37,103 ECGs collected at the University of Tokyo Hospital. LV systolic dysfunction was assessed by echocardiography and defined as an ejection fraction of less than 40%. Patients who had LV systolic dysfunction were labeled as positive, and the rest were labeled as negative. Table 1 shows the details of the dataset. The dataset was divided into five splits: *train1*, *valid1*, *train2*, *valid2*, and *test*. It is worth noting that, to avoid data leakage, all ECGs from one patient were assigned to the same split. The *train1* and *valid1* splits were used for self-supervised pretraining, in which the labels were not used. The *train2*, *valid2*, and *test* splits were used for the downstream task, in which a CNN model was trained to detect LV systolic dysfunction by supervised learning. Due to the imbalance of the labels, all positive samples were used in the downstream dataset, which means only the negative samples were used for self-supervised pretraining.

Self-Supervised Pretraining: The first step of our approach was to pretrain an efficient encoder by self-supervised contrastive learning, using the *train1* and *valid1* splits. In this study, we used the MoCo [4, 5] algorithm. In the MoCo algorithm, data transformation methods should be used, and the main idea is that feature vectors generated from the same sample should be close to each other, and those generated from different samples should be far apart. More details of MoCo are showed in Appendix A. Inspired by previous works [12, 13], we used five transformation methods: *Gaussian noise*, *Gaussian smoothing*, *random erasing*, *resizing*, and *baseline shifting*. Figure 1 shows examples of each method. While training, several transformation methods were randomly selected to apply to the ECG data, and the transformed data were then input into the encoder.

Model Architecture: In most previous studies, a ResNet-based [17] architecture was used for the encoder. However, in this study, we used a CNN architecture, as shown in Appendix B. The reason is that a previous study showed that this architecture can achieve high performance in detecting LV systolic dysfunction [11]. The encoder consists of six temporal convolution blocks, one spatial convolution block, and one fully connected layer. The encoder takes a 12×5000 ECG matrix as input, and finally outputs a 128-dimensional feature vector. In the self-supervised pretraining, ECGs (after normalization and transformation were applied) were input to the encoder to perform contrastive learning, using MoCo. In the downstream task, two fully connected layers and a sigmoid layer were added to the encoder to detect LV systolic dysfunction. The binary cross-entropy loss was minimized.

Table 1: Details of the dataset used in this study.

	Split	Negative	Positive	Total	(Unique Patients)	Labels Used
Pretraining	train1	20693	0	20693	(11324)	no
	valid1	5303	0	5303	(2864)	no
Downstream	train2	5406	2427	7833	(3347)	yes
	valid2	1108	514	1622	(685)	yes
	test	1092	560	1652	(699)	yes
Total		33602	3501	37103	(18919)	

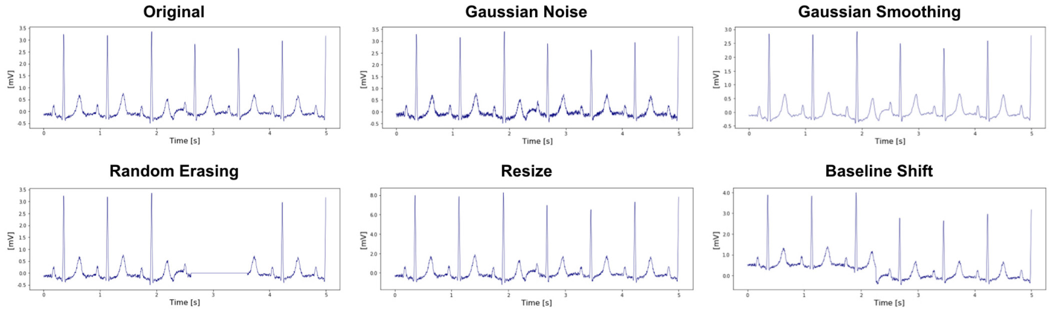


Figure 1: Single-lead examples of each transformation method used to train the MoCo encoder.

Experiments and Baselines: We conducted three experiments: our proposed approach, *baseline (unpretrained)*, and *baseline (full labels)*. Our proposed approach first trained an encoder using self-supervised contrastive learning on the pretraining dataset (*train1* and *valid1*), and then fine-tuned the model on the downstream dataset (*train2* and *valid2*) using the pretrained encoder. *Baseline (unpretrained)* omitted the self-supervised pretraining step. It directly trained a model on the downstream dataset using supervised learning, with the model weight initialized randomly. *Baseline (full labels)* also directly performed supervised learning. However, in contrast with the *baseline (unpretrained)*, it used all labels in the dataset, including the labels of the *train1* and *valid1* splits. The *train1* and *train2* splits were concatenated as the training dataset, and the *valid1* and *valid2* splits were concatenated as the validation dataset. Table 3 in Appendix C shows the correspondence of the data splits used in each experiment. In all three experiments, the Adam optimizer was used. Initial learning rates of 1e-4, 3e-5, and 1e-5 were tested in each experiment; the model weights with the lowest validation loss were saved, and later used for evaluation on the test set.

3 Results and Discussion

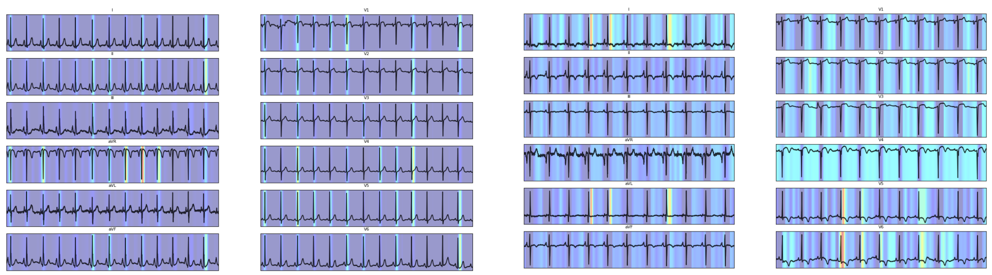
The area under the curve (AUC) of each model, evaluated on the *test* split, is shown in Table 2. Our proposed approach achieved an AUC of 0.9245 and it outperformed both the *baseline (unpretrained)* and *baseline (full labels)*, of which the AUC values were 0.9208 and 0.9236, respectively. The fact that our proposed approach performs better than the *baseline (unpretrained)* indicates that self-supervised pretraining enables the encoder to learn effective feature representations, which are useful for detecting LV systolic dysfunction. Moreover, our proposed approach achieved higher performance than the *baseline (full labels)*. The *baseline (full labels)* used all labels in the dataset; the total number of labels used in the *train1* and *train2* splits is 28,526. In contrast, our proposed approach used only 7,833 labels in the *train2* split. In summary, our proposed approach achieved higher performance than the *baseline (full labels)* using only 28% of the labels.

To understand the features learned by the self-supervised pretraining, we used Grad-CAM [16] to visualize the area of focus of the pretrained encoder in the ECG, as shown in Figure 2. The figure shows that, in the negative sample, the encoder focuses on the QRS complex. In contrast, in the positive sample, the encoder focuses on the ST segment, in particular, in leads I, aVL, V5, and V6.

Although our approach enables us to develop a deep learning model to detect LV systolic dysfunction in a label-efficient way, because of the limited number of data in the test dataset, it is hard to show statistically significant differences between our approach and the baselines. Future work should evaluate our approach on a larger dataset. Moreover, an exciting direction for future work is to explore the best model architecture and data transformation methods for self-supervised contrastive learning for electrocardiography data.

Table 2: AUC of each model evaluated on the test dataset.

	Baseline (unpretrained)	Baseline (full labels)	**Proposed**
AUC	0.9208	0.9236	**0.9245**



(a) Negative sample

(b) Positive sample

Figure 2: Two examples of Grad-CAM visualization of the pretrained encoder.

4 Broader Impact

Our study used self-supervised pretraining to advance the performance of a deep learning model to detect LV systolic dysfunction. Compared with the conventional supervised learning approach, the number of labels required was significantly reduced. The most obvious potential practical benefit of our work is that it may enable the development of a high-performance model to detect cardiac diseases using only a small amount of labeled data. Our work will accelerate the application of self-supervised learning to the analysis of ECGs. It will also contribute to the clinical understanding of ECGs, which may potentially lead to future work with greater clinical impact. To the best of our knowledge, our work has no potential negative impacts.

References

[1] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for AI. *Communications of the ACM*, 64(7):58–65, 2021.

[2] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1), 2021.

[3] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[8] Zachi I Attia, Peter A Noseworthy, Francisco Lopez-Jimenez, Samuel J Asirvatham, Abhishek J Deshmukh, Bernard J Gersh, Rickey E Carter, Xiaoxi Yao, Alejandro A Rabinstein, Brad J Erickson, et al. An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm: a retrospective analysis of outcome prediction. *The Lancet*, 394(10201):861–867, 2019.

[9] Joon-myoung Kwon, Kyung-Hee Kim, Zeynettin Akkus, Ki-Hyun Jeon, Jinsik Park, and Byung-Hee Oh. Artificial intelligence for detecting mitral regurgitation using electrocardiography. *Journal of electrocardiology*, 59:151–157, 2020.

[10] Joon-Myoung Kwon, Soo Youn Lee, Ki-Hyun Jeon, Yeha Lee, Kyung-Hee Kim, Jinsik Park, Byung-Hee Oh, and Myong-Mook Lee. Deep learning–based algorithm for detecting aortic stenosis using electrocardiography. *Journal of the American Heart Association*, 9(7):e014717, 2020.

[11] Zachi I Attia, Suraj Kapa, Francisco Lopez-Jimenez, Paul M McKie, Dorothy J Ladewig, Gaurav Satam, Patricia A Pellikka, Maurice Enriquez-Sarano, Peter A Noseworthy, Thomas M Munger, et al. Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram. *Nature medicine*, 25(1):70–74, 2019.

[12] Han Liu, Zhenbo Zhao, and Qiang She. Self-supervised ECG pre-training. *Biomedical Signal Processing and Control*, 70:103010, 2021.

[13] Temesgen Mehari and Nils Strodthoff. Self-supervised representation learning from 12-lead ECG data. *arXiv preprint arXiv:2103.12676*, 2021.

[14] Pritam Sarkar and Ali Etemad. Self-supervised ECG Representation Learning for Emotion Recognition. *IEEE Transactions on Affective Computing*, pages 1–1, 2020.

[15] Dani Kiyasseh, Tingting Zhu, and David A Clifton. CLOCS: Contrastive Learning of Cardiac Signals Across Space, Time, and Patients. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5606–5615. PMLR, 2021.

[16] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-Cam: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Appendix

A Framework of the MoCo Algorithm

In the MoCo algorithm, data transformation methods should be used to generate positive and negative sample pairs. The query ECG and key ECG are respectively transformed and then input to the encoder. The feature vectors of each ECG are used to calculate the contrastive loss. By minimizing the contrastive loss, the feature vectors generated from the same sample will become close to each other, and those generated from different samples will be far apart.

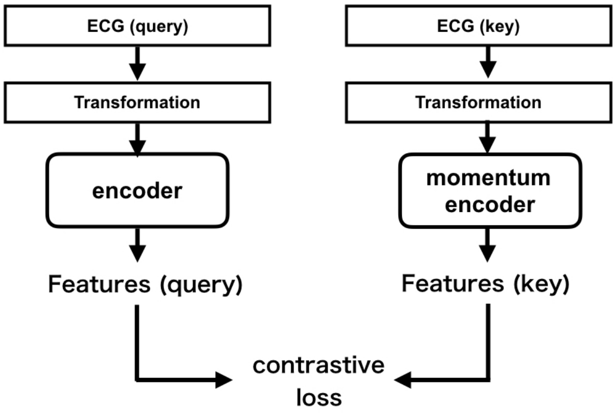


Figure 3: Framework of the MoCo algorithm.

B Model Architecture of the MoCo Encoder

The encoder consists of six temporal convolution blocks, one spatial convolution block, and one fully connected layer. As proposed in the previous work [11], the temporal convolution blocks were designed to learn features within each lead. Each block has a 2D convolutional layer with filters in the shape of $1 \times K$ to perform convolution in the time direction of each lead. The spatial convolution block has a 2D convolutional layer with filters in the shape of 12×1 to perform convolution in the channel direction.

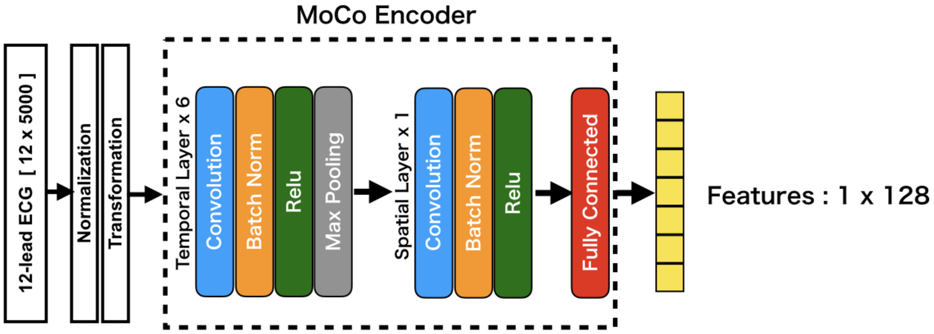


Figure 4: Model architecture of the MoCo encoder.

C Data Splits Used in Each Experiment

Table 3: Data splits used in the proposed method and the baselines. The bracketed number indicates the number of labels used.

	Pretrain		Downstream		
	Training	Validation	Training	Validation	Evaluation
Proposed	train1 (0)	valid1 (0)	train2 (7833)	valid2 (1622)	test2 (1652)
Baseline (unpretrained)	-	-	train2 (7833)	valid2 (1622)	test2 (1652)
Baseline (full labels)	-	-	train1+train2 (28526)	valid1+valid2 (6952)	test2 (1652)