Adversarial Masking for Pretraining ECG Data Improves Downstream Model Generalizability

Anonymous Author(s)
Affiliation
Address
email

Abstract

Medical datasets often face the problem of data scarcity, as ground truth labels must be generated by medical professionals. One mitigation strategy is to pretrain deep learning models on large, unlabelled datasets with self-supervised learning (SSL), but this introduces the issue of domain shift if the pretraining and task dataset distributions differ. Data augmentations are essential for improving the generalizability of SSL-pretrained models, but they tend to be either handcrafted or randomly applied. We use an adversarial model to generate masks as augmentations for 12-lead electrocardiogram (ECG) data, where masks learn to occlude diagnostically-relevant regions [1]. Compared to random augmentations, adversarial masking reaches better accuracy on a downstream arrhythmia classification task under a domain shift condition and in data-scarce regimes. Adversarial masking is competitive with, and even reaches further improvements when combined with state-of-art ECG augmentation methods, *3KG* and *random lead masking*.

1 Introduction

Across medical applications, deep learning is increasingly used to automate disease diagnosis Miotto et al. [2018]. In some cases, neural networks have even reached or exceeded the performance of expert physicians [Hannun et al., 2019]. One such application is with 12-lead electrocardiogram (ECG) data, which is commonly collected to screen for various cardiovascular disorders [Fesmire et al., 1998]. There has been a recent surge in ECG-based deep learning research, largely enabled by challenges like the annual PhysioNet/Computing in Cardiology Challenge [Perez Alday et al., Reyna et al., 2021]. While these research outcomes show substantial progress within the field, their performance only reflects training on large-scale, labelled dataset. In contrast, real world medical datasets are likely much smaller due to the extensive resources required to collect medical labels.

Data scarcity is a well-documented issues that effect deep learning training. Models trained on small datasets lack generalizability to unseen data and cannot be deployed reliably [Kelly et al., 2019]. To mitigate issues associated with small training datasets, large yet unlabelled dataset can be leveraged to pretrain deep learning models with robust representations, which is commonly done in the ECG domain [Sarkar and Etemad, 2020, Weimann and Conrad, 2021, Liu et al., 2021, Kiyasseh et al., 2021, Diamant et al., 2022, Mehari and Strodthoff, 2022, Oh et al., 2022]. However, if the two datasets are collected under different environments, with different sensors, or across different populations, the transferrability of the model to the downstream task can be greatly impacted [Koh et al., 2021].

Contrastive self-supervised learning (SSL) is a pretraining technique that does not require a labelled dataset and can induce robust representations in the model Chen et al. [2020a]. The encoding model is trained to maximize the similarity between latent representations of augmented pairs of data,

[1] Code available at: `https://anonymous.4open.science/r/advmask_ecg/`

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

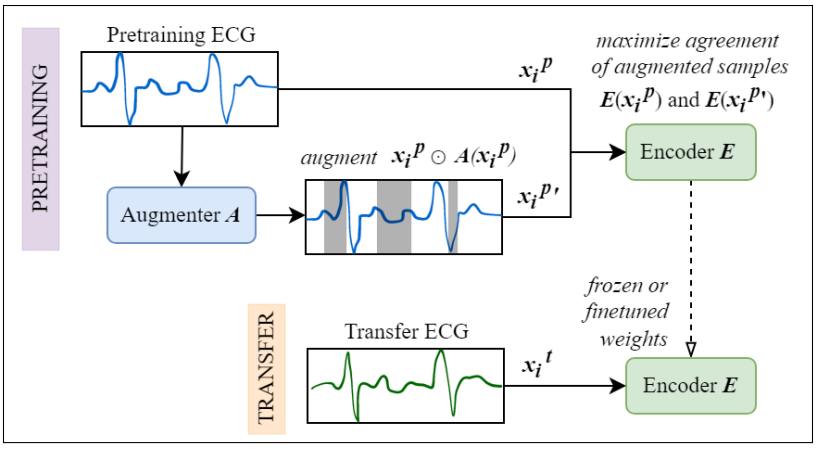

Figure 1: Adversarial masking pretraining and downstream transfer phases.

while separating from the representations of other data samples. SLL-pretrained models learn the underlying structure of the data invariant to the augmentations, which means that selecting good augmentations is crucial for learning useful representations. State-of-art within the ECG domain is patient-level contrastive SSL (PL-SSL), where pairs of data are taken from the same patient at different points in time Kiyasseh et al. [2021], Diamant et al. [2022]. The performance of PL-SSL techniques are powerful, but can be improved in combination with other augmentations.

For time-series data, augmentations are typically selected from a standard pool like noise injection, baseline shift, time-domain masking, and more [Wen et al., 2021]. A recent work *3KG* develops physiologically-consistent spatial augmentations for ECGs that reach good performance on small datasets [Gopal et al., 2021]. Another introduces *random lead masking (RLM)*, where leads of the ECG are fully masked out at random Oh et al. [2022]. However, such augmentations are either handcrafted with manually tuned parameters per dataset or randomly applied, while we investigate the use of an adversarial method to optimize augmentation parameters.

In this work, we elect to focus on time-domain masking, a type of augmentation with high potential but is underexploited for periodic time-series data like ECGs:

- We adapt the image-based adversarial masking method of Shi et al. [2022] to generate masks as augmentations for each given ECG sample during PL-SSL pretraining. To our knowledge, this is the first work to implement adversarial masking for time-series data.
- We show that incorporating adversarial masking improves performance in a downstream heart arrhythmia classification task compared to baseline augmentations. We also find orthogonal benefits when combined with state-of-art ECG augmentations, *3KG* and *RLM*.

2 Methods

The overview of the adversarial masking pretraining framework with the encoding model $\boldsymbol{E}$ and adversarial masking model $\boldsymbol{A}$ and the downstream transfer learning step is show in Figure 1.

2.1 Self-Supervised Pretraining

CMSC Objective: We use Contrastive Multi-segment Coding (CMSC) as the PL-SSL strategy from Kiyasseh et al. [2021]. Given a batch of data $\left\{\boldsymbol{x} \in \mathbb{R}^D\right\}_{i=1}^B$, the positive pair is created as $\boldsymbol{x}_i = \boldsymbol{x}^{1..\frac{D}{2}}, \boldsymbol{x'}_i = \boldsymbol{x}^{\frac{D}{2}..D}$, where $\boldsymbol{x}_i, \boldsymbol{x'}_i$ represent temporally non-overlapping ECG segments from the same patient. We further transform a data sample $\boldsymbol{x'}_i = T(\boldsymbol{x'}_i)$, where T represents one or more augmentations. The encoding model $\boldsymbol{E}$ is trained to align the feature representations of the positive pair of data, $\boldsymbol{h}_i = \boldsymbol{E}(\boldsymbol{x}_i)$ and $\boldsymbol{h'}_i = \boldsymbol{E}(\boldsymbol{x'}_i)$, while separating them from all other negative samples within the batch where $\boldsymbol{x}_i \neq \boldsymbol{x}_j$. We perform training with the SimCLR objective [Chen et al., 2020b], described in Equation 1, where $\tau = 0.1$ is a temperature scaling term.

$$\mathcal{L}_{\text{SimCLR}}(\boldsymbol{x}; \boldsymbol{E}) = \log \frac{\exp(sim(\boldsymbol{h}_i, \boldsymbol{h'}_i)/\tau)}{\sum_{i \neq j} \exp(sim\left(\boldsymbol{h}_i, \boldsymbol{h'}_j\right)/\tau)} \tag{1}$$

Adversarial Objective: We add an adversarial masking model $\boldsymbol{A}$ that generates a set of *N* masks for a given data sample $\boldsymbol{m}_i \in \mathcal{R}^{NxD} = \boldsymbol{A}(\boldsymbol{x}_i)$. Having multiple masks ensures that the masked regions of the ECG are alternated, as only one mask is sampled and applied in each training step. The masking model $\boldsymbol{A}$ acts in opposition to the encoding model $\boldsymbol{E}$ by generating difficult augmentations, hence maximizing the SimCLR objective. We also adapt the sparse penalty from Shi et al. [2022] to limit the amount of masking, described by Equation 2 with $\alpha = 0.1$ as a weighting term. The full *min-max* loss objective for $\boldsymbol{E}$ and $\boldsymbol{A}$ is described by Equation 3:

$$\mathcal{L}_{\text{sparse}}(\boldsymbol{x}; \mathcal{A}) = \alpha \sin\left(\frac{\pi}{D}\sum_{d=1}^{D} \boldsymbol{m}_d\right)^{-1} \tag{2}$$

$$\min_{\mathcal{E}} \max_{\mathcal{A}} \mathcal{L}_{\text{SSL}}(\mathcal{E}, \mathcal{A}) - \mathcal{L}_{\text{sparse}}(\mathcal{A}) \tag{3}$$

Architectures: The encoder backbone is a 1D ResNet-18 [He et al., 2016], the best performing architecture in multiple ECG diagnostic tasks [Nonaka and Seita, 2021]. The masking model is a 1D U-Net [Ronneberger et al., 2015] with four downsampling and upsampling layers. If the number of masks $N = 1$, the output function is a *sigmoid*. If $N > 1$, the output function is *softmax*, which enforces that each mask covers different regions. A mask is randomly sampled, binarized, and applied with probability *p = 0.8* to the corresponding sample of the batch. Further details are in Appendix A.

2.2 Downstream Transfer Learning

We transfer the encoding models's learned representations to a downstream classification task and measure the classification accuracy as the benchmarking metric. In *linear evaluation*, the pretrained encoding model's weights are frozen and a linear output layer is trained on the classification task. In *finetuning*, the weights of both the encoding model and the linear layer are trained and updated. We use a standard cross-entropy loss for training.

3 Results

3.1 Transfer Task Performance

CMSC pretraining is performed with 12-lead ECG datasets from the PhysioNet/Computing in Cardiology Challenge 2020 [Perez Alday et al.], which comprises over 66 thousand patient recordings from six institutions in four countries. The downstream transfer task is arrhythmia classification with the 12-lead Chapman-Shaoxing ECG dataset, which has over 10 thousand patients and four classes of cardiac rhythm labels [Zheng et al., 2020]. There are no overlaps between the pretraining and downstream datasets, so our transfer learning setup constitutes a *domain shift*.

We present transfer learning results with *linear evaluation* and *finetuning* transfer conditions using a train-validation-test split of 80%, 20% and 20%. We simulate real world data scarcity conditions by reducing the transfer training dataset to 100%, 10%, and 1% of the original size (8516 samples). Table 1 shows the prediction accuracy and standard deviation of the arrhythmia classification task with models pretrained on all baseline and adversarial augmentations (details of the baseline augmentation in Appendix B). The size of the test set is held constant despite the reduced training datasets and the test accuracy for all experiments is reported as the average across 3 random seeds. The best performing results are in bold. Results from a *Scratch* baseline with a random encoding model is also reported. Adversarial masking (**Adv Mask**) results are reported with $N = 2$, which performs the best.

Adversarial masking yields superior results to most baseline augmentations across all training dataset sizes, including other random masking techniques. It is competitive to but does not consistently outperform *3KG* [Gopal et al., 2021] and *RLM* [Oh et al., 2022], two methods which are specific to ECGs and encourages learning invariances across leads. However, **Adv Mask** combined with *3KG*

Table 1: Downstream classification accuracy across multiple dataset sizes reported for encoders pretrained on multiple augmentations and a *Scratch* baseline, using CMSC pretraining strategy.

	100% Training Dataset		*10% Training Dataset*		*1% Training Dataset*	
	Linear	Finetune	Linear	Finetune	Linear	Finetune
Scratch	49.88 ± 1.87	70.99 ± 1.59	38.77 ± 4.16	63.92 ± 1.97	22.31 ± 0.12	34.89 ± 0.54
Gaussian	79.82 ± 0.481	78.04 ± 0.61	71.06 ± 0.98	70.34 ± 1.11	59.82 ± 0.29	60.39 ± 1.48
Powerline	79.07 ± 2.07	78.22 ± 1.3	69.84 ± 1.88	69.68 ± 2.31	58.73 ± 1.04	58.92 ± 0.79
STFT	79.6 ± 1.37	78.16 ± 1.37	70.62 ± 1.01	70.27 ± 1.48	60.92 ± 1.32	60.64 ± 2.74
Wander	79.41 ± 0.65	77.94 ± 0.55	71.12 ± 0.33	70.74 ± 0.24	59.61 ± 1.37	59.39 ± 1.39
Shift	80.44 ± 1.58	79.44 ± 1.67	70.74 ± 1.12	70.81 ± 1.52	60.04 ± 2.26	59.73 ± 1.34
Mask	72.58 ± 4.55	72.18 ± 2.44	68.54 ± 4.29	68.03 ± 3.48	54.85 ± 4.86	53.57 ± 3.39
Blockmask	84.01 ± 2.31	82.85 ± 2.12	74.22 ± 0.79	74.25 ± 0.84	64.61 ± 1.14	63.17 ± 0.90
3KG	91.55 ± 0.93	91.74 ± 0.78	86.2 ± 1.05	86.58 ± 1.33	75.72 ± 1.03	76.44 ± 1.08
RLM	90.58 ± 0.12	91.86 ± 1.42	87.48 ± 0.56	87.98 ± 0.67	76.03 ± 1.16	76.94 ± 0.65
Adv Mask (AM)	90.11 ± 0.38	88.96 ± 0.32	86.05 ± 1.58	85.58 ± 2.24	76.66 ± 3.59	75.88 ± 4.14
AM + 3KG	92.87 ± 1.08	92.65 ± 0.85	90.58 ± 1.27	90.8 ± 1.19	82.54 ± 1.94	81.88 ± 1.54
AM + RLM	**93.27** ± **0.77**	**93.21** ± **0.50**	**91.24** ± **0.76**	**91.18** ± **0.48**	**82.76** ± **1.12**	**83.92** ± **0.66**

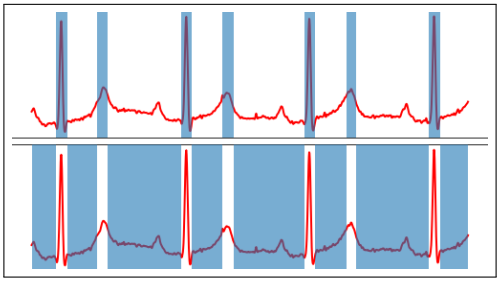

Figure 2: N=2 adversarially generated masks overlaying Lead II of the ECG sample.

and *RLM* shows significant improvements over the respective augmentations on their own, with **Adv Mask + RLM** reaching the best performance in all transfer trials. This shows that benefits introduced by adversarial masking is orthogonal to other augmentations and consistent in data scarcity.

3.2 Analysis of Augmentations

We visualize the generated masks in Figure 2, with more examples in Appendix C. Lead II of the ECG is displayed in red and overlaid with blue masked regions (the values of which are set to 0 in training). With $N = 2$, one mask consistently covers the QRS complex and T-wave, while the other mask covers the remaining areas. We hypothesize this is similar to capturing diagnostically-relevant "semantic" content of the ECG, in the same way that Shi et al. [2022] demonstrates adversarial image masks cover semantically-coherent regions of the image. According to an analysis of the salient regions of ECG data, the QRS complex often carry high levels of disease-diagnostic information [Jones et al., 2020]. This suggests that the encoding model emphasizes learning the structual information of the highly salient regions of the ECGs during pretraining.

4 Conclusions and Future Work

We show that meaningful masking for ECG data can be utilized positively in SSL pretraining of deep learning models. Such models learn generalizable representations for downstream transfer learning in highly data-scarce and domain-shifted tasks. Furthermore, adversarial masking as an augmentation scheme is agnostic to the choice of architecture and training technique, so the method can be extended to other SSL frameworks and to other time-series data modalities. Future work includes evaluating the types of downstream tasks that adversarial masking brings the most benefit to.

References

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020*, pages 699–708, 2020a. URL `https://proceedings.mlr.press/v149/nonaka21a.html`.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020b. URL `https://proceedings.mlr.press/v119/chen20j.html`.

Nathaniel Diamant, Erik Reinertsen, Steven Song, Aaron D. Aguirre, Collin M. Stultz, and Puneet Batra. Patient contrastive learning: A performant, expressive, and practical approach to electrocardiogram modeling. *PLoS Computational Biology*, 18(2):1–16, 2022.

F. M. Fesmire, R. F. Percy, J. B. Bardoner, D. R. Wharton, and F. B. Calhoun. Usefulness of Automated Serial 12-Lead ECG Monitoring During the Initial Emergency Department Evaluation of Patients With Chest Pain. *Annals of Emergency Medicine*, 31(1):3–11, 1998. ISSN 0196-0644. doi: 10.1016/S0196-0644(98)70274-4.

Bryan Gopal, Ryan Han, Gautham Raghupathi, Andrew Ng, Geoff Tison, and Pranav Rajpurkar. 3KG: Contrastive Learning of 12-Lead Electrocardiograms using Physiologically-Inspired Augmentations. *Proceedings of Machine Learning Research*, 158:156–167, 2021. URL `https://proceedings.mlr.press/v158/gopal21a.html`.

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 2019 25:1*, 25(1): 65–69, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Yola Jones, Fani Deligianni, and Jeff Dalton. Improving ECG Classification Interpretability using Saliency Maps. In *IEEE 20th International Conference on Bioinformatics and Bioengineering*, pages 675–682. Institute of Electrical and Electronics Engineers Inc., 2020. doi: 10.1109/BIBE50027.2020.00114.

Christopher J. Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, 17(195): 1–9, 2019. doi: 10.1186/S12916-019-1426-2.

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, PMLR*, pages 5606–5615. PMLR, 2021. URL `https://proceedings.mlr.press/v139/kiyasseh21a.html`.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

Han Liu, Zhenbo Zhao, and Qiang She. Self-supervised ECG pre-training. *Biomedical Signal Processing and Control*, 70:103010, 2021. doi: 10.1016/J.BSPC.2021.103010.

Temesgen Mehari and Nils Strodthoff. Self-supervised representation learning from 12-lead ECG data. *Computers in Biology and Medicine*, 141:105114, 2022. doi: 10.1016/j.compbiomed.2021.105114.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T. Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6):1236–1246, 2018. doi: 10.1093/BIB/BBX044.

Naoki Nonaka and Jun Seita. In-depth Benchmarking of Deep Neural Network Architectures for ECG Diagnosis. In *6th Machine Learning for Healthcare Conference*, volume 149 of *Proceedings of Machine Learning Research*, pages 414–439. PMLR, 2021. URL https://proceedings.mlr.press/v149/nonaka21a.html.

Jungwoo Oh, Hyunseung Chung, Joon-myoung Kwon, Dong-gyun Hong, and Edward Choi. Lead-agnostic Self-supervised Learning for Local and Global Representations of Electrocardiogram. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 338–353. PMLR, 2022. URL https://proceedings.mlr.press/v174/oh22a.html.

Erick A Perez Alday, Annie Gu, Amit Shah, Chad Robichaux, An Kwok, Ian Wong, Chengyu Liu, Feifei Liu, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, Qiao Li, Ashish Sharma, Gari D Clifford, and Matthew A Reyna. Classification of 12-lead ECGs: the PhysioNet/ Computing in Cardiology Challenge 2020.

Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Lauren Milechin, Julia Mullen, Andrew Prout, Antonio Rosa, Charles Yee, and Peter Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2018.

Matthew A. Reyna, Nadi Sadr, Erick A.Perez Alday, Annie Gu, Amit J. Shah, Chad Robichaux, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, Sardar Ansari, Hamid Ghanbari, Qiao Li, Ashish Sharma, and Gari D. Clifford. Will Two Do? Varying Dimensions in Electrocardiography: The PhysioNet/Computing in Cardiology Challenge 2021. *Computing in Cardiology*, 48:1–4, 2021. doi: 10.23919/CINC53138.2021.9662687.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention – MICCAI 2015*, 9351:234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Pritam Sarkar and Ali Etemad. Self-supervised ECG Representation Learning for Emotion Recognition. *IEEE Transactions on Affective Computing*, pages 1–13, 2020. doi: 10.1109/TAFFC.2020.3014842.

Yuge Shi, N Siddharth, Philip H S Torr, and Adam R Kosiorek. Adversarial Masking for Self-Supervised Learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20026–20040. PMLR, 2022. URL https://proceedings.mlr.press/v162/shi22d.html.

Kuba Weimann and Tim O.F. Conrad. Transfer learning for ECG classification. *Scientific Reports*, 11 (1):1–12, 2021. doi: 10.1038/s41598-021-84374-8.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time Series Data Augmentation for Deep Learning: A Survey. *IJCAI International Joint Conference on Artificial Intelligence*, pages 4653–4660, 2021. doi: 10.24963/ijcai.2021/631.

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), 2020. doi: 10.1038/s41597-020-0386-x.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We mention limitation of baseline experiments in Appendix B.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Not within scope of extended abstract

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Link in footnote of abstract (first page).
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details in Appendix A and B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Standard deviations are provided across 3 seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details in Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Specified in codebase or references.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Not discussed, but datasets used are open source and anonymized.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Architecture and Training

All models are implemented with PyTorch Lightning. Training and testing are performed with a single NVIDIA Volta V100 GPU on the MIT Supercloud Reuther et al. [2018].

Encoding Model: We adapt the ResNet-18 architecture used by Nonaka and Seita [2021] in their benchmarking study, which had superior performance out of eight backbones. The hidden dimension is 512. We also use a two-layer projection head typical in the contrastive SSL framework [Chen et al., 2020b] to convert the encoder's outputs to a 128-dimension space. This detail was omitted from the main paper for simplicity and the projector is not transferred to the downstream task.

Adversarial Model: We adapt an open-source 1D U-Net implementation with four downsampling layers, each layer comprising three consecutive *Conv1d* and *BatchNorm* blocks and two *Linear* layers with *ReLU* activation. A shallower version of the U-Net with three downsampling layers was also tested, but it yielded poorer results. The hidden representation is upsampled with four *Upsample* and *Conv1D* blocks. The last layer is a *Conv1D* layer with the number of out channels being the number of masks N. The outputs are passed through either a *softmax* or *sigmoid* function depending on N.

Pretraining Conditions: In pretraining, we use a learning rate of 0.0001 and an *Adam* optimizer for both the encoding model and adversarial model. The models are updated one after another with their respective losses within the same batch. The batch size is 32 with gradient accumulation over 4 batches, which is equivalent to an effective batch size of 128. To reduce computation efforts, we also use mixed precision training. An early stopping condition based on minimizing $\mathcal{L}_{\text{SSL}}$ is implemented to lower training time — typically training terminates in 20-40 epochs.

As mentioned in the main text, *N=2* was found to consistently achieve the best results, as we believe it is due to the masked regions being alternated during training. To overcome potential issues with catastrophic forgetting due to overactive masking of the training data, we only apply the sampled mask with a probability of *p=0.8*.

As the pretraining dataset is labelled, we also attempted to incorporate the classification accuracy of the pretraining dataset into the *min-max* objective, as described in the modified objective in Equation 4. However, this had a slight negative impact the transfer performance so its results are not reported.

$$\min_{\mathcal{E}} \max_{\mathcal{A}} \mathcal{L}_{\text{SSL}}(\mathcal{E}, \mathcal{A}) + \mathcal{L}_{\text{classification}}(\mathcal{E}) - \mathcal{L}_{\text{sparse}}(\mathcal{A}) \tag{4}$$

Transfer Conditions: Transfer learning is performed with a batch size of 256 (or the full dataset in the case of the 1% training dataset size trials), learning rate of 0.01, and an *Adam* optimizer. A single *Linear* layer projects the output dimension of the encoding model to the number of output classes.

B Augmentations

The baseline augmentations are sourced from ECG deep learning research. While the list is not exhaustive, we aim to be representative of typical random ECG augmentation techniques used in existing works. Note that a limitation of our results is that we only test with isolated augmentations, whereas combining two or more augmentations may intuitively yield better results.

Gaussian Noise: A vector of noise $v(t) \sim \mathcal{N}(0, 0.05)$ is sampled and added to each lead of the ECG. Gaussian noise injection is a very common augmentation used in SSL training for both time-series and image data.

Powerline Noise: Powerline noise $n(t) = \alpha cos(2\pi t k f_p + \phi)$, with $\alpha \sim \mathcal{U}(0, 0.5)$, $\phi \sim \mathcal{N}(0, 2\pi)$, and $f_p = 50Hz$ is added to each lead of the ECG [Mehari and Strodthoff, 2022, Oh et al., 2022].

Short-time Fourier Transform: STFT involves computing the Fourier transform of short segments of the time-series signal to generate a spectrogram. A random mask with values sampled from a beta distribution $B(\alpha = 5, \beta = 2)$ is applied to the spectrogram. Finally, the STFT operation is inversed to recover the time domain signal.

Baseline Wander: The ECG signal is perturbed with a very low frequency signal to simulate drifting: $n(t) = C \sum_{k=1}^{K} a \cos(2\pi t k \Delta f + \phi)$, with $C \sim \mathcal{N}(1, 0.5^2)$, $\alpha \sim \mathcal{U}(0, 0.5)$, $\phi \sim \mathcal{N}(0, 2\pi)$, and $\Delta f \sim \mathcal{U}(0.01, 0.2)$ [Mehari and Strodthoff, 2022, Oh et al., 2022].

Baseline Shift: A fraction $p = 0.2$ of the baseline of each lead of the ECG signal is shifted positively or negatively by a factor of $\alpha \sim \mathcal{N}(-0.5, 0.5)$ [Mehari and Strodthoff, 2022, Oh et al., 2022].

Mask: Two types of random masking are implemented, where Mask refers to any timestep being masked out to 0 with the probability $p = 0.2$.

Blockmask: Blockmask masks out a continuous portion $p = 0.2$ of each lead to 0. The main difference is that Blockmask would occlude larger structural regions of the ECG, whereas Mask only occludes local details.

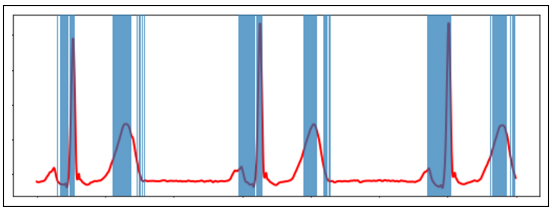

Figure 3: *N*=1 adversarially generated masks overlaying Lead I of the ECG sample.

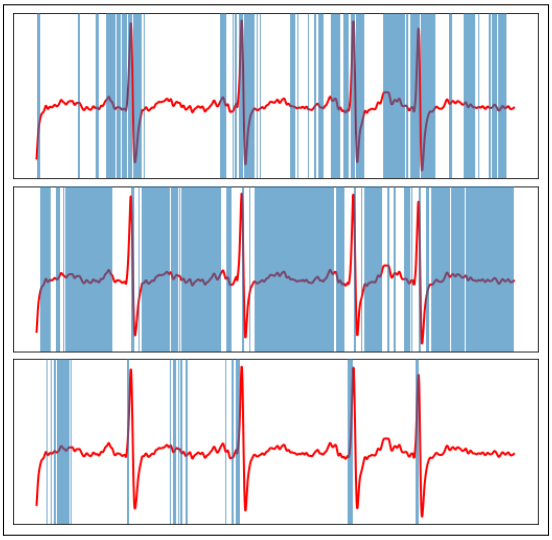

Figure 4: *N*=3 adversarially generated masks overlaying Lead I of the ECG sample.

3KG: Developed by Gopal et al. [2021], 3KG augments the 3D spatial representation of the ECG in vectorcardiogram (VCG) space and reprojects it back into ECG space, mimicking natural variations in cardiac structure and orientation. We take the best parameters reported in the paper, a random rotation $-45° \leq \theta \leq 45°$ and a scaling factor $1 \leq s \leq 1.5$.

RLM: Introduced by Oh et al. [2022], RLM fully occludes individual ECG leads with probability *p=0.5*, which was the parameter used in their paper. This reduces the dependency on requiring all 12 leads to extract useful information.

C Adversarial Masks

Figures 3-5 depict more examples of adversarially generated masks with $N = 1, 3, 12$. When $N < 12$, one mask is randomly sampled and applied to all leads of the ECG. When $N = 12$, each mask is applied separately to each lead. As mentioned in the paper, $N \neq 2$ trials saw reduced performance so their results are not reported.

What should be noted is that the adversarial masking technique is extremely good at identifying peaks in the ECG (primarily the QRS complex and T-wave). Increasing *N* does not seem to be helpful as masks tend to cover either peaks or flat regions, which can be achieved with *N*=2. When there are *N*=12 masks, we note that the same regions of each lead are masked, showing that lead-specific masks are redundant.

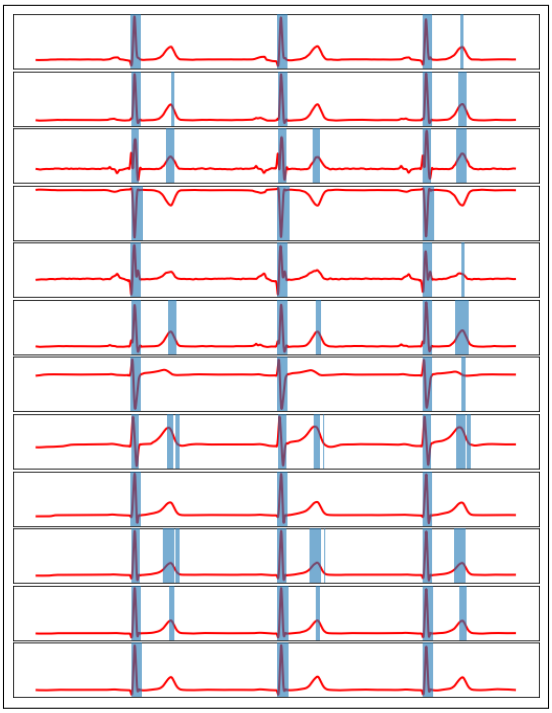

Figure 5: N=12 adversarially generated masks overlaying Lead I of the ECG sample.