Proceedings of CONF-SPML 2027 Symposium: Intelligent Network Security and Machine Learning for Communication Systems
DOI: 10.54254/2755-2721/2027.MELB37869

Supervised Contrastive Learning for Cross-Load Bearing Fault Diagnosis with Limited Training Labels

Haoran Li

New Canaan High School, New Canaan, USA
leoli090730@gmail.com

Abstract. Vibration-based fault diagnosis requires annotated examples that may not be easy to come by in service. This paper compares the effectiveness of supervised contrastive pretraining to cross-entropy training using the same compact convolutional encoder and 40 recordings of bearings from Case Western Reserve University. Motor load data with 0 and 1 hp load provided the training set, 2 hp provided the validation data, and 3 hp the test set. Five random seeds were evaluated at three training budgets of 5, 10 and 20 examples per class. A separate fixed validation set with 944 labeled windows was used to select checkpoints and was excluded from the budgets above. Linear classification on contrastively pretrained embeddings resulted in higher mean test accuracy and macro-F1 score compared to cross-entropy training at all three budgets. The biggest improvement came at a budget of 10 windows/class: accuracy increased from 84.81% to 89.22% and macro-F1 score from 84.62% to 89.24%. Recall for ball faults was primarily responsible for the increase. These results support the effectiveness of complete contrastive training based on this cross-load test scheme but do not isolate the effect of the loss or establish performance on other machines.

Keywords: rolling bearing, fault diagnosis, limited labels, supervised contrastive learning, vibration signal

1. Introduction

Damage to bearings results in impacts as rolling elements traverse a defect. Thus, vibration signals can assist in locating the existence and position of any fault [1]. One such benchmark is the Case Western Reserve University (CWRU) that has been used in many studies despite being different from faults in practice [2]. Machine learning techniques replace manually designed features with automatically learned ones [3]. Deep learning models facilitate this process [4], and one-dimensional convolutional neural networks (1D-CNNs) can learn features directly from the waveform [5]. Labeled data is required during training of these models, which makes it difficult for a small dataset to fully cover all possible fault cases and conditions. Transfer learning solves the problem by using a related dataset [6]. Contrastive learning represents yet another approach that could leverage the available observations. SimCLR is a framework that uses transformations to learn representations of individual data samples [7]. In turn, supervised contrastive learning (SupCon) relies both on class labels, and on the assumption that different views of the samples within the same class form a

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2027.MELB37869

positive pair [8]. This family of methods has been studied for the task of diagnosing the incipient bearing faults [9], few-shot learning of high-frequency time series [10] and diagnosis with limited annotations [11]. This current work investigates the performance of the SupCon protocol with a view to assessing whether the application of the SupCon protocol leads to better performance compared to direct training with cross-entropy loss with the same small encoder, as well as the evolution of the difference for the three label budgets. The novelty of the current paper lies in the comparative approach and not in proposing any new methods or architectures. The matching sampling protocol, test set motor load, and per-class error serve as the foundation for the results' interpretation.

2. Related Work

2.1. Bearing vibration and diagnostic evaluation

The classical approach of bearing diagnostics connects vibrations to fault frequencies, structural resonances, and operating speed. With the help of filtering and envelopes, repetitive fault-related impacts hidden in the raw signal can be revealed [1]. Some statistical properties like root-mean-square amplitude and kurtosis can be used to describe other properties of the signal. This is good physical insight, but the most valuable features depend on the measurement process.

With 1D-CNN, the local filters are learned automatically rather than defined beforehand [4, 5]. Having shared convolutional weights limits the number of parameters while enabling the detection of the same pattern at various positions within the window. Classification of waveforms still does not prove that the fault mechanisms were learned by the network. Besides, it might use some peculiarities of a particular recording or operating mode.

The above makes a difference in the case of assessing CWRU data. Windows within a single recording have common characteristics of load, speed, fault and measurement configuration, and random splitting into training and testing can benefit from the identification of this common context. Moreover, the benchmark includes unusual signatures that make for a non-textbook interpretation [2]. Splitting at the level of individual records minimizes reuse of recording-related information.

2.2. Representation learning with limited labels

The type of data available determines what limited-label techniques can be utilized. Zheng et al. use transfer learning to bring information from a bigger database of bearings to a smaller database [6]. In such a scenario, there is extra information that would not be available to a model that only sees the target data. When comparing results, one must consider not only the size of the labeled dataset but also any other extra data source.

SupCon makes sure that the anchor embedding is similar to other embeddings within the same category while dissimilar to other embeddings in other categories [8]. Such a loss function may come in handy in cases where the classes of faults have impulsive signal patterns that are not easily separable. However, its efficacy will depend on how the views are considered equivalent.

Ding et al. perform self-supervised pre-training with unlabeled data for bearing fault detection [9]. Chen et al. study supervised contrastive few-shot learning for high-frequency time series, along with the construction of augmentation views [10]. Kong et al. perform representation transfer from unlabeled vibration data to diagnose downstream tasks [11]. Relevant literature precedents have been created here. In this experiment, there is no pre-training performed using any external dataset with a supervised objective, hence not a comparison against all of these works.

DOI: 10.54254/2755-2721/2027.MELB37869

2.3. Scope of the comparison

Training was performed in two ways with respect to the encoder with the encoder architecture, selected training windows and evaluation sets held constant. In the first case, cross-entropy was used to train both a classifier and an encoder together. The second option included augmentations, supervised contrastive learning of the encoder, and classification using a linear model based on the frozen features. There were differences between the augmentation and training periods, therefore, all success should be attributed to the entire process.

3. Methodology

3.1. Data and cross-load split

Forty acceleration recordings of the drive-end side were chosen from the official CWRU data [12]. The rig employs a 2 hp motor with a dynamometer. Single point defects were made by means of the electrical-discharge machining technique. The set includes four normal recordings and 36 faulty recordings: inner race, ball and outer race faults with the diameters of 0.007, 0.014 and 0.021 in. and 0, 1, 2 and 3 hp loading. The outer race faults used the 6 o'clock position. Four classes were defined as normal, inner race, ball and outer race conditions, fault diameter was not a class.

The 0 and 1 hp load levels served as the training pool, the 2 hp load level as the validation set, and the 3 hp load level as the test set. Assignment was done prior to window extraction. No single source file was used for more than one split. The training pool had 3,421 windows. The validation and test sets were randomly sampled without replacement into 236 and 237 windows, respectively, per class with constant seeds 20260818 and 20260819. The resulting 944 validation windows and 948 test windows were identical across multiple runs. This holdout corresponds to the motor load rather than independent bearing specimens.

3.2. Signal windows and normalization

The 48 kHz signals were partitioned into non-overlapping 2,048 sample windows, each having an approximate duration of 42.7 ms. Trailing windows that were incomplete were excluded. Normalization of each window was done based on the individual window's mean and population standard deviation:

$$\tilde{x} = \frac{x - \mu_x}{\sigma_x + 10^{-6}} \tag{1}$$

In equation (1), x refers to the input window, while μx and σx denote its mean and standard deviation, respectively. Normalization makes the model insensitive to offset and scaling but loses the absolute amplitude value. The windows were sampled randomly at each budget, distributing the class budget as equally as possible among its corresponding data source records. The same seed was used to sample the same windows for both approaches. The budgets were sampled independently rather than as nested subsets. The number of windows per class (5, 10 and 20 windows) is calculated only on the training set, the 944 windows used for selecting checkpoints are extra.

DOI: 10.54254/2755-2721/2027.MELB37869

3.3. Shared convolutional encoder

The encoder was comprised of three convolution layers, having numbers of channels 1→16→32→64. Convolution kernel sizes were 64, 16, and 8, stride values were 8, 4, and 2, while the zero-padding was 28, 6, and 3. All convolutions were accompanied by group normalization of the 4, 8, and 8 groups, respectively, as well as a rectified linear unit activation function. Finally, adaptive average pooling was used to form the 64-dimension embedding. The number of parameters of the encoder was 25,936.

The baseline included an extra linear layer with four outputs, making up 26,196 parameters in total. Adam was employed to minimize cross-entropy for 100 epochs with a learning rate of 0.001 and a weight decay of 0.0001. The batch size was set as the minimum of 32 and the number of training windows. After epoch 1 and each fifth epoch, validation macro-F1 was calculated to pick up the best checkpoint that achieved the maximum value. Test data was not included in this procedure.

3.4. Supervised contrastive training

The contrastive encoder applied 64→64→32 projection head with a rectified linear unit in between and unit-length output normalization. In each window, two augmented views were created. Each was subjected to a random circular shift of up to 64 data points and Gaussian noise with standard deviation that was drawn randomly from 0.01 to 0.04. With 50% probability, an interval of 32 to 128 points would be turned into zero. These augmentations were expected to leave the label intact, they were not checked for physical validity.

For an anchor i, let z_i be its normalized projection, and P(i) be all other views in the batch that have the same label, including the second view of the same window. The per-anchor loss with temperature τ is [8]

$$L_i = -\frac{1}{|P(i)|}\sum_{p\in P(i)} log\left(\frac{exp(z_i\cdot z_p/\tau)}{\sum_{a\neq i} exp(z_i\cdot z_a/\tau)}\right) \quad (2)$$

The denominator contains all the batch examples except the anchor, while the loss during training is the average of equation (2) for the anchors. During pretraining, τ = 0.1, Adam, for 200 epochs, was employed with a learning rate of 0.001, weight decay of 0.0001, and batch sizes of 64 original windows at most. The encoder network was frozen, and the projection layer was dropped. Training of the 260-parameter linear classifier was done on the embeddings without augmentation for 200 epochs using Adam, a learning rate of 0.005, a weight decay of 0.0001, and batch sizes of 32 windows at most. The classifier checkpoint followed the baseline's validation rule, the pretrained encoder was taken from the final pretraining epoch.

3.5. Evaluation

Accuracy and macro-F1 were computed on the test set with fixed load of 3 hp. Accuracy is the fraction of correct classifications. Class F1 is 2TP/ (2TP + FP + FN), while macro-F1 is the average of all classes' F1 values. Recall is TP/ (TP + FN). Row-normalized confusion matrices show which classes affect aggregate differences.

There were five seeds per method and budget, resulting in 30 CPU experiments performed. Seeds were the same as the draws for training samples, not the entire sequence of stochastic steps in the two approaches. Summary means and sample standard deviations show the five repetitions, pair-wise differences were also computed. Deterministic PyTorch operations were used. This accounts

DOI: 10.54254/2755-2721/2027.MELB37869

for sensitivity to sampling and training randomness on one fixed split, not machine variance. No statistical significance test was conducted.

4. Results and discussion

4.1. Performance across label budgets

In the case of 10 training windows per class, the contrastive method showed 89.22% accuracy and 89.24% macro-F1, as opposed to 84.81% and 84.62% achieved by cross-entropy (Table 1, Figure 1). In all the five matching experiments, SupCon won. The mean difference between paired macro-F1 was 4.62%, with the sample standard deviation being 2.74. It is the highest mean advantage seen in experiments for three different budgets.

For five training windows per class, the mean difference between macro-F1 was 2.42 ± 6.38 percentage points, calculated without rounding. In that case, mean macro-F1 increased from 80.72% to 83.15%, with three wins in five paired runs. For twenty training windows per class, the respective means were 86.55% and 88.31%, with the paired difference being 1.76 ± 2.51 percentage points and four wins out of five.

Table 1. Cross-load test performance

Windowsper class	1D-CNNAccuracy (%)	SupConAccuracy (%)	1D-CNNMacro-F1 (%)	SupConMacro-F1 (%)
5	81.31 ± 2.10	83.48 ± 3.91	80.72 ± 2.22	83.15 ± 4.28
10	84.81 ± 1.14	89.22 ± 2.22	84.62 ± 1.34	89.24 ± 2.20
20	86.62 ± 1.44	88.16 ± 1.20	86.55 ± 1.62	88.31 ± 1.09

Note: Values are mean ± sample standard deviation across five seeds, in percent. Training budgets exclude 944 labeled validation windows.

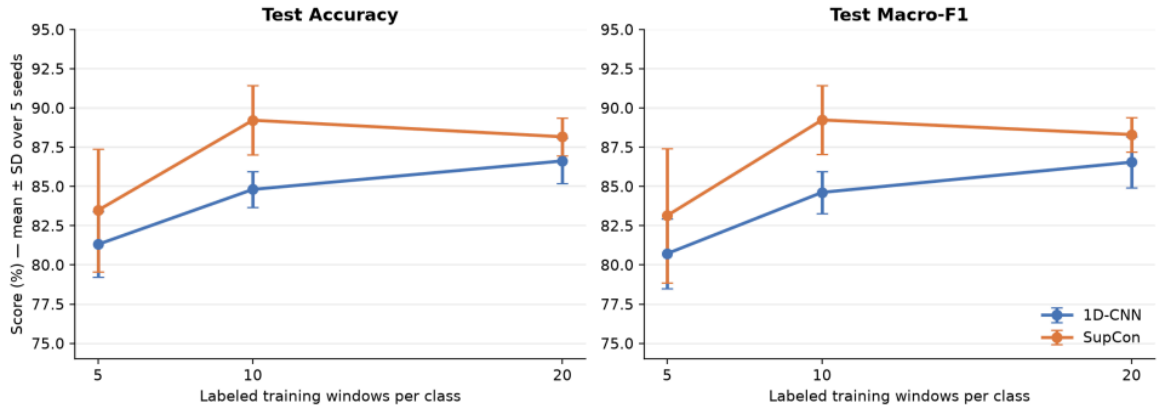


Figure 1. Test accuracy and macro-F1 versus labeled training windows per class

Note: Points show means and error bars ±1 sample standard deviation across five runs. The validation and test sets are fixed.

4.2. Changes in class-specific errors

Normal windows were always correctly recognized using both approaches in all runs. At 10 and 20 windows per class, recall of the baseline inner race faults was also very high and varied from 95.11% to 95.53%. The majority of mistakes were made while recognizing ball and outer-race faults

DOI: 10.54254/2755-2721/2027.MELB37869

since there were several windows with balls included in the inner-race class (Figure 2). Overall improvements can thus be considered according to those specific to each type of fault.

The recall of ball fault at 10 windows/class increased from 71.39% to 87.43%, and the recall of outer-race fault increased from 72.32% to 76.62%. Recall of the inner-race fault decreased from 95.53% to 92.83%. In case of 20 windows, the ball recall improved from 80.76% to 91.98%, whereas the inner-race recall decreased from 95.11% to 88.27%. The outer-race recall changed slightly, from 70.63% to 72.41%. SupCon did not increase recall for all classes.

The increase in recognition of the balls is consistent with the hypothesis about the new representation allowing separating overlapping patterns better. However, the confusion matrixes do not prove this statement. Embedding distance measurements and controlled augmentations are required to prove this fact.

4.3. Instability with five labels per class

SupCon macro-F1 scores across five training windows per class varied from 76.28% to 86.95%. Mean gain in this respect thus masks a significant chance of a poor run. Since each fault class draws from six source recordings, five training windows cannot account for all of them. What waveform and severity values are present may be important, but no attempt was made to disentangle this effect from randomization in initialization and augmentation.

The baseline performance rose with the budget from 10 to 20 training windows, while the SupCon mean macro-F1 dropped from 89.24% to 88.31%. Both trends reduced the difference. That alone is insufficient evidence that 10 windows is the optimal budget, in fact, only three budgets were used and samples were not nested, and training hyperparameters were held constantly. The evidence does not confirm the hypothesis that fewer labels imply greater contrastive advantage.

For pretraining, two views, a projection head, and a more extensive schedule are necessary. After training, both approaches have 26,196 parameters, since the projection head is discarded. Equality in number of parameters does not imply equality in runtime and suitability for an edge device. These were not measured.

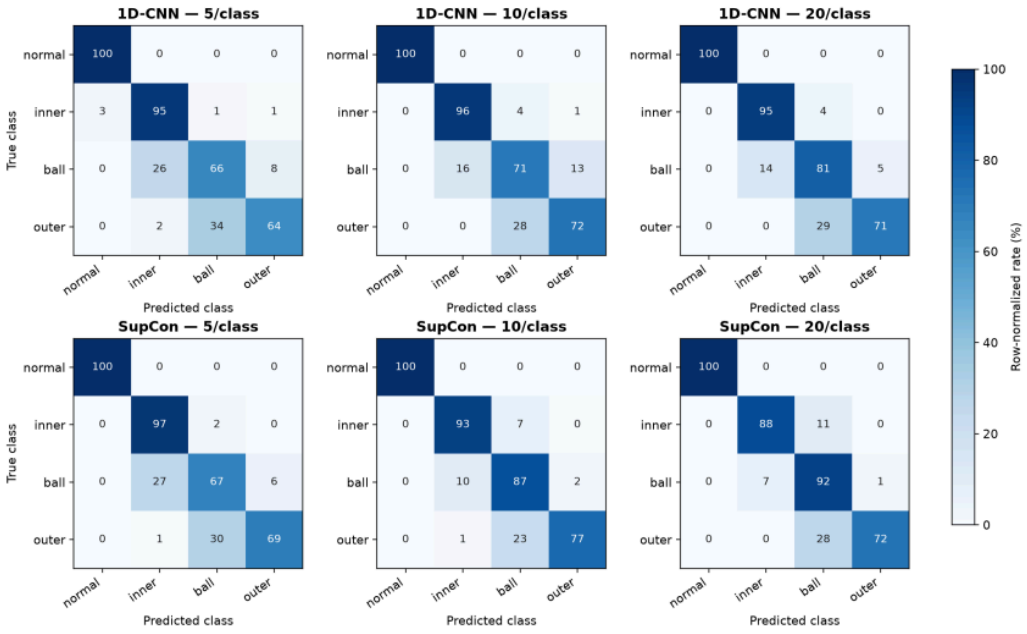


Figure 2. Test confusion matrices pooled across five seeds and normalized by true class

DOI: 10.54254/2755-2721/2027.MELB37869

Note: Rows denote true classes and columns predictions, entries are percentages. The repeated predictions use the same 948 windows, not independent test samples.

4.4. Limitations of the evidence

The strongest qualification has to do with the scarcity issue. Training windows are limited in number, while the choice of checkpoints is able to utilize 944 validation windows that have been labeled. This means that the current study cannot claim to have diagnosed based on only 20–80 labels at all. The CWRU labels are part of the source condition: a lot of windows from the same record do not amount to having independent labels of the bearing.

The benchmark of a balanced and seeded-fault database does not simulate the development of damage nor the proportion of normal operating conditions that happens in reality. Optimization time was not equal, therefore, the loss-only justification could not happen. Loading three hp makes for one split, and there are only five seeds for generalization. The augmentation of cross-entropy control is needed for a better comparison.

5. Conclusion

The study also contrasted cross-entropy training with supervised contrastive pretraining for four-class bearing classification. In both cases, the same small 1D convolutional encoder architecture and identical window sizes, based on the CWRU dataset, were used. Records were split by motor load prior to window extraction to avoid having a single record both in training and testing, resulting in a cross-load evaluation instead of evaluating a model on a new machine.

In terms of accuracy and macro-F1, SupCon outperformed cross-entropy training on average at all three levels of budget. The most apparent difference was at 10 windows per class, where accuracy and macro-F1 were boosted from 84.81% and 84.62%, respectively, to 89.22% and 89.24%. In that setting, all five repetitions preferred SupCon. At five windows per class, the average macro-F1 increase was 2.42%, but the individual runs were significantly less consistent. Insufficient amount of data itself could not ensure improvement.

This gap narrowed again at 20 windows, with the baseline improving and SupCon's mean score dropping. Results from the classes provide additional context. With 10 windows in each class, ball fault recall increased by 16.04%, whereas inner race recall decreased by 2.70 points. Thus, an improvement in the overall performance masked an unfavorable balance between fault classes. The number of parameters used did not change due to the deletion of the contrastive projection head before classification.

These results do not establish that the contrastive loss is independently superior, this study only highlights that further research on the full contrastive process is justified. Large amounts of annotated data in the validation set, different training processes, and identical laboratory settings are limitations to this conclusion. The next comparison must ensure the equality of training, augmentation, effort and validation labels. It will help to understand if the improvements achieved are sustainable when the model selection information and the training environment change.

References

[1] Randall, R. B., & Antoni, J. (2011). Rolling element bearing diagnostics - A tutorial. *Mechanical Systems and Signal Processing, 25*(2), 485-520.

[2] Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mechanical Systems and Signal Processing, 64-65*, 100-131.

DOI: 10.54254/2755-2721/2027.MELB37869

[3] Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing, 138*, 106587.
[4] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing, 115*, 213-237.
[5] Li, H., Zhang, Q., Qin, X., & Sun, Y. (2020). Raw vibration signal pattern recognition with automatic hyper-parameter-optimized convolutional neural network for bearing fault diagnosis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 234*(1), 343-360.
[6] Zheng, Z., Fu, J., Lu, C., & Zhu, Y. (2021). Research on rolling bearing fault diagnosis of small dataset based on a new optimal transfer learning network. *Measurement, 177*, 109285.
[7] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *Proceedings of Machine Learning Research, 119*, 1597–1607.
[8] Khosla, P., Teterwak, P., et al. (2020). Supervised contrastive learning. *Advances in Neural Information Processing Systems, 33*, 18661–18673.
[9] Ding, Y., Zhuang, J., Ding, P., & Jia, M. (2022). Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. *Reliability Engineering & System Safety, 218*, 108126.
[10] Chen, X., Ge, C., Wang, M., & Wang, J. (2023). Supervised contrastive few-shot learning for high-frequency time series. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(6), 7069–7077.
[11] Kong, D., Zhao, L., Huang, X., Huang, W., Ding, J., Yao, Y., Xu, L., Yang, P., & Yang, G. (2023). Self-supervised knowledge mining from unlabeled data for bearing fault diagnosis under limited annotations. *Measurement, 220*, 113387.
[12] Case Western Reserve University. (2026). Bearing Data Center. *Bearing Data Center*. https://engineering.case.edu/bearingdatacenter. Accessed September 13.