Good Fences Make Good Neighbours

Anonymous ICCV submission

Paper ID ****

Abstract

Neighbour contrastive learning enhances the common contrastive learning methods by introducing neighbour representations to the training of pretext tasks. These algorithms are highly dependent on the retrieved neighbours and therefore require careful neighbour extraction in order to avoid learning irrelevant representations. Potential "Bad" Neighbours in contrastive tasks introduce representations that are less informative and, consequently, hold back the capacity of the model making it less useful as a good prior. In this work, we present a simple yet effective neighbour contrastive SSL framework, called "Mending Neighbours" which identifies potential bad neighbours and replaces them with a novel augmented representation called "Bridge Points". The Bridge Points are generated in the latent space by interpolating the neighbour and query representations in a completely unsupervised way. We highlight that by careful selection and replacement of neighbours, the model learns better representations. Our proposed method outperforms the most popular neighbour contrastive approach, NNCLR, on three different benchmark datasets in the linear evaluation downstream task. Finally, we perform an in-depth three-fold analysis (quantitative, qualitative and ablation) to further support the importance of proper neighbour selection in contrastive learning algorithms.

1. Introduction

Deep Learning (DL) algorithms have made remarkable strides across a wide range of applications [13]. The success of DL can be attributed to larger architectures, powerful computation capabilities and more importantly, the availability of large training data [2]. Collecting large volumes of labelled data is often expensive, time-consuming, and very scarce in many domains [40]. Self-supervised Learning (SSL) is an alternative learning paradigm that enables models to learn meaningful representations by exploiting massive raw data without annotated supervision [16]. SSL models are label agnostic and learn representations that are generic across several tasks [1]. They capture the underlying relationships, structure or semantics of the data using a pretext task [37]. Downstream tasks based on the pretext trained models are therefore able to perform better on fine-tuning using task-specific labels [25, 32, 41]. Well-designed pretext tasks which learn proper representations rather than free-style learning would be better priors in various downstream tasks.

Figure 1. **Sample images** showing "good" and "bad" neighbours.

Pretext tasks can be classified in general into generative, contrastive or generative contrastive [29]. Generative models use an encoder-decoder architecture to reconstruct the sample [22, 23, 36]. Contrastive Learning (CL) algorithms, on the other hand, work on pulling together different augmentations (views) of the sample closer (positives) to each other while repelling those from other instances (negatives) [21]. CL algorithms use several similarity measurements such as NCE Loss [18], InfoNCE loss [33], and Redundancy-reduction loss [44] to contrast different views. SimCLR [8], a breakthrough SSL method used two views of the same image to learn the visual representations. MoCo [19] extended SimCLR by using a dynamic queue to store representations of views. Self-distillation methods such as BYOL [17], SimSiam [9], and DINO [6, 34] rely on different encoders to map the different views to each other. Other methods such as SWaV [5] and Barlow Twins [44] use correlation to infer relationships between views.

One of the fundamental design criteria in the CL algorithms is the generation of positive views from a given sample [1]. Data augmentation serves as a common approach to generate different diverse views from the sample image. SSL methods learn by contrasting the different views to learn representations that are invariant to these transformations [4]. However, there is a potential pitfall in solely using data augmentation to create different views. The augmentations alone would not be able to cover all variations of a given class [14].

Neighbour Contrastive Learning (NCL) algorithms are based on the notion that data augmentations (views) may not provide sufficient diverse information in selecting positive samples [14]. NCL algorithms contrast different views of the image with their nearest neighbours and learn to bring them in close proximity. This allows for better-learned representations as the contrasted pairs are often from different source samples. Algorithms such as NNCLR [14], MeanShift [24], All4One [15] are able to learn from new data points that would be different from those generated using views. SNCLR [10] used cross-attention to compute the importance of neighbours and used them as soft neighbours. A common entity in these methods is that they use a support set (queue) to store the representations of samples and use algorithms such as k-nearest neighbours [14] or mean shift [24] to retrieve one or few nearest neighbours, which in turn act as positive samples during CL.

One of the critical aspects for the proper functioning of these algorithms lies in the careful selection of neighbours [14]. Fig. 1 shows some examples of ”good” and ”bad” neighbours. ”Good” neighbours are essential to learn proper representations of data distribution as they share similar features. Good representations possess local smoothness, sparse activation for specific inputs, temporal and spatial coherence, hierarchically organized explanatory factors, and simple dependencies [3]. ”Good” neighbours do not need to be from the same semantic class, rather should produce representative features. ”Bad” neighbours, on the other hand, may introduce noise or confusion in the representations that might lead to less effective representations. It is therefore crucial to identify good neighbours that can positively help SSL models to learn proper representations of the data. With this aim, we explore the question of *What constitutes a good neighbour?* We propose a neighbour correction framework that identifies potential ”bad”, *not so* helpful neighbours and uses the identified neighbour representations to generate new synthetic representations that are effective and also different from the representations created using different views of the samples.

The main contributions of our work are characterized as follows: (1) We present a neighbour correction framework through which we identify potential ”bad” neighbours that can harm the pretext training process. (2) We introduce a mechanism to generate representation in the latent space, called ”Bridge Points” from those identified neighbours such that they move closer to the instances in the latent space. (3) With a detailed analysis of the performance of our method on different benchmarks, we show the importance of neighbour selection in CL frameworks.

2. Related Works

In this section, we present an overview of the latest self-supervised visual representation learning literature that is relevant to our work.

Self-supervised Learning. SSL involves training a model without using any kind of supervised signal in an attempt to force it to learn intermediate representations that could be later transferred to multiple downstream tasks [1]. Existing SSL methods can be grouped into generative and discriminative algorithms [29]. While the former requires the use of visual transformers and reconstruction tasks, the latter has been able to maintain good results with a low budget thanks to their CL pretext tasks [46, 34]. CL works on grouping similar samples closer and moving diverse samples farther from each other [21]. In the context of learning image representations, the objective function relies on positive pairs, where both representations belong to the same semantic class and negative pairs, consisting of representations from different semantic classes. The goal is to bring the positive pairs closer together in the feature space, while simultaneously pushing away the negative pairs to avoid the collapse of the model. In recent years, this principle has been leveraged into several alternatives that work on clusters [5], using only positive samples [17] using neighbours as positives [14, 24, 10, 15]. Neighbour-based algorithms are characterized by their enhanced generalization capacity inherited from the use of diverse neighbour representations obtained by algorithms such as k-NN.

Neighbour Contrast Approaches. Nearest neighbour (NN) is a simple and effective machine learning algorithm applied in several computer vision tasks [6, 35, 39]. NN-based SSL methods leverage the relationships between samples in the pretext training to enhance the quality of the learned representations. By exploiting the proximity or similarity between samples, these methods encourage the model to capture meaningful patterns, structures, or semantics from the data. NNCLR [14] was the first SSL method that explicitly adopted the NN approach. NNCLR implemented a memory queue, called a support set, to store the representations of samples and contrasted representations between views of samples and their first nearest neighbour mined from this support set. Mean Shift [24] used a mean-shift algorithm to group several neighbours together without contrasting them directly. SNCLR [10] used a cross-attention module to measure the correlation between samples and used this score to identify positive samples in CL. Recently, All4One [15] combined the neighbour contrast with feature contrast and transformer-based centroid contrast to learn representations from different latent spaces. The core idea in all the above-listed approaches is exploiting neighbours to learn relationships between samples. However, they do not control or measure the neighbours extracted, which could lead to a performance decrease. Our work differentiates from them by emphasizing the importance of a good neighbour selection and proposes useful replacements for the ones that should be discarded.

Feature Space Augmentations. Image data augmentations play a critical role in supervised learning [45, 42, 20, 38] and in most of CL-SSL approaches [8, 44, 17, 10, 15]. Creating two different samples from the exact same initial sample allowed unsupervised CL, as no labels are required for the correct selection of the contrasted samples [8]. Several pipelines have been proposed in order to enhance the augmented sample and their usefulness [17, 4]. All these augmentations are directly applied to the images so when augmentations are required for latent representations, it is not really effective. On the contrary, latent space augmentations can be perfectly applied with negligible computational efficiency loss. Adding random Gaussian noise, and extrapolating or interpolating feature space representations are the most common approaches to create new augmented representations [12, 7]. In recent years, these kinds of augmentations have been used to address diverse problems such as long-tailed instance segmentation [43], pose prediction [28] and multimodality [30]. However, the lack of visual control has made latent augmentations less popular than their counterpart. In our work, we propose a novel application of these latent augmentations in an NCL task in order to create interpolated representations. These interpolations, when contrasted, improve the capabilities of the trained model by enabling the model to capture more discriminative and meaningful patterns. This way, they provide meaningful replacements for neighbours where the extracted ones do not provide useful information for the CL task.

3. Rationale

NNCLR [14], which marked the inception of NCL algorithms, proved that changing from augmented representations increased the diversity of contrasted samples and, consequently, improved the performance of models on several downstream tasks. However, NNCLR also showed that a semi-supervised selection of neighbours achieved better results compared to an unsupervised selection. This highlights the fact that not all neighbours are completely useful. NCL algorithms often use a k-NN to extract the nearest neighbours of samples by computing the distances in latent space. These neighbours are later used in learning meaningful representations of the data. High-quality neighbour representations, therefore, directly impact the performance of the trained models [14]. Improving the quality of neighbour extractions in an unsupervised manner poses several challenges. Identifying what constitutes a *"good neighbour representation"* is not straightforward. All NCL models compute their neighbours in feature space, making their analysis more difficult. Additionally, there are no direct measures of quality between neighbour representations. Moreover, it is very challenging to differentiate an augmentation of a sample from its neighbour representation. Considering these complexities, we rise several important questions in this work: "*Can an image augmentation be a neighbour?*", "*How do we measure the quality of a neighbour in CL?*", "*How do we identify potential bad neighbours in feature space?*" and, most importantly, "*What does constitute a good neighbour?*". In this work, we hypothesize that good neighbours are those that are different from data augmentations, but in close proximity to the samples, whereas bad neighbours are those that are farthest from the augmentations. Based on this hypothesis, we provide "Mending Neighbours", a method based on neighbour selection and replacement neighbour generation.

4. Mending Neighbours

The foundations of our proposed pipeline are established with inspirations from the NNCLR [14] framework. The proposed pipeline is shown in Fig. 2. It contains two *braches*, each composed by an encoder followed by an MLP projector, together defined as f. One of the branches has an additional MLP predictor. Each f transforms the input image into an SSL representation. For a given mini-batch $\mathcal{X}$, we augment the samples twice, one for each branch to obtain the augmented batches $\mathcal{X}^1$ and $\mathcal{X}^2$. These batches are passed through their respective branches to obtain their respective representations $\mathcal{Z}^1$ and $\mathcal{Z}^2$. On every iteration, z_i is extracted from its respective representation batch and is used as a query for the k-NN algorithm that extracts its neighbour representation nn_i from a fixed-sized Support Set, $\mathcal{Q}$.

We extract the neighbours following NNCLR [14], which is defined as follows:

$$\mathcal{NN}(z_i, \mathcal{Q}) = argmin\,(Sim(z_i, \mathcal{Q})) \tag{1}$$

where $Sim(z_i, \mathcal{Q})$ is defined as $||z_i - \mathcal{Q}||_2$. Next, we use a simple neighbour evaluation approach to identify the "good" and the potential "bad" neighbours. We evaluate the goodness, gd_i of each neighbour by storing the similarity between the query sample and its nearest neighbour representation in the feature space. This is defined as:

$$\mathcal{G}(z_i, \mathcal{Q}) = min\,(Sim(z_i, \mathcal{Q})) \tag{2}$$

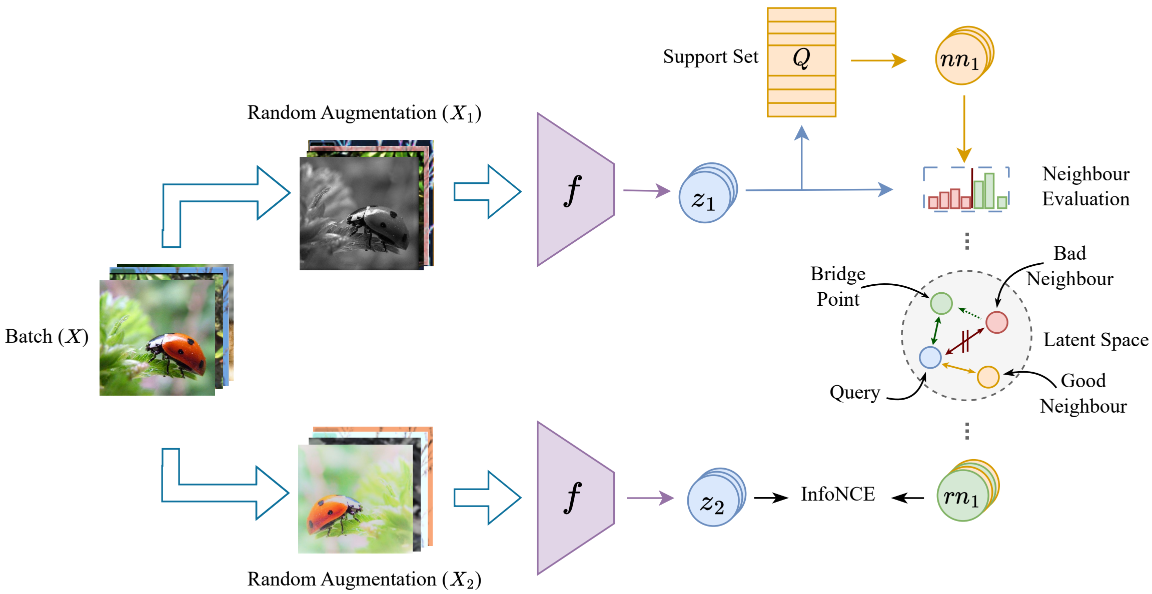

Figure 2. **Overview of our proposed method.**

We use the mean gd_i of the whole batch as a threshold to split the neighbours into "good" and potential "bad" ones.

For the identified as "bad" neighbours, we present an unsupervised feature space interpolation between the "bad" neighbour nn_i and the query sample z_i. This interpolation allows us to create representations in the feature space that has the characteristics of both the query and the neighbours. We augment the potential "bad" neighbour directly in the feature space by creating an interpolation or Bridge Point (BP) bp_i between neighbour representation and its query. Though the identified neighbour can deteriorate the learning of the model, they still would contain representative information as they are the most similar in the Support Set. Formally, the interpolation is defined as:

$$bp_i = (z_i - nn_i) * \lambda + nn_i \tag{3}$$

λ is used to control the strength of the interpolation.

The final neighbour replacement function $\mathcal{R}$ is defined as follows:

$$\mathcal{R}(z_i, nn_i, bp_i) = \left\{ \begin{array}{ll} nn_i, & \text{if } gd_i > \frac{1}{n}\sum_{k=1}^{n} gd_k \\ bp_i, & otherwise \end{array} \right\} \tag{4}$$

This approach aims to detect the "bad" neighbours while also replacing them with representations created in the feature space between the query and the "bad" neighbours. The final loss is determined as:

$$\mathcal{L}_i = -log\left(\frac{exp(rn_i^1 \cdot z_i^2/\tau)}{\sum_{k=1}^{N} exp(rn_i^1 \cdot z_k^2/\tau)}\right) \tag{5}$$

where rn_i^1 represents the output of the defined replacement function $\mathcal{R}$ and τ is the temperature constant. The loss is computed symmetrically.

5. Validation

In this section, we first show the experimental settings of our proposed framework and then present our results highlighting the need to use "good" neighbours in NCL. We use three popular image classification benchmarks: CIFAR-10 [26], CIFAR-100 [26], and ImageNet-100, a reduced ImageNet of 100 classes [27] to validate our method. We compare our proposed method to the NCL SoA, specially to the benchmark NCL algorithm NNCLR [14].

5.1. Implementation Details

For all datasets, we use a ResNet-18 encoder in a self-supervised manner. We use solo-learn [11], a Pytorch-based SSL framework for all our implementations. Regarding the architecture, we follow the implementations of NNCLR [14] and use a common shared-weights dual encoder-projector architecture with a predictor at the end of the second branch. We create the projectors using 3 fully-connected layers of size [2048, 2048, 256] and the predictor using 2 fully-connected layers of size [4096, 256]. All fully-connected layers are followed by batch normalization. For all experiments, we initialize the backbones with solo-learn initialization parameters [11]. We follow the hyper-parameter settings as defined by solo-learn for all datasets except for the queue size of CIFAR experiments, where we increase it to 98304 following NNCLR [14]. We empirically set the interpolation hyperparameter λ to 0.2 for CIFAR10 and ImageNet100 datasets and 0.5 for CIFAR100. We train all the models using a single NVIDIA RTX 3090 GPU.

5.2. Results

We analyse the benefits of our proposed approach using linear evaluation on the three benchmark datasets, following common SSL evaluation schemes. We further present quantitative results based on neighbour retrieval and similarity metrics. We also show visual qualitative results highlighting the advantages of having "good" neighbours in NCL.

5.2.1 Linear Evaluation

For linear evaluation, we freeze the SSL-trained models and use them as backbones or feature extractors for a common linear classification task. Following the solo-learn pipeline [11], we perform the linear evaluation across all training epochs and report the best Top-1 accuracy. We present the linear evaluation results in Table 1. Our "Mending neighbour" approach outperforms NNCLR [14] on the three benchmarks showing the advantages of having a smarter selection and replacement of neighbours. Selection of good neighbours leads to better-learned models that act as a better prior in the linear classification task. One interesting point to note is that the datasets with a high number of classes show a bigger improvement in terms of performance. This could be due to the fact that the higher the number of classes, the easier for the k-NN to fail in retrieving a "good" neighbour as more confusing classes could be present in the support set. However, this is not the case for CIFAR-10, which has a total of 10 well-differentiated classes.

5.2.2 Quantitative Results

In addition to the linear evaluation, we also analyse the accuracy of the neighbours extracted for both NNCLR and the proposed approach using the k-NN accuracy. We show the k-NN accuracy for both CIFAR datasets in Table 2. This measures the number of times the extracted neighbour belongs to the same class as that of the query. As can be seen in Table 2, our approach increases the retrieval accuracy of the neighbours in both cases, implying that the generated bridge point representations of the encoder contain higher representative information than those obtained using the NNCLR neighbours.

We also measure the similarity or goodness of the extracted neighbours for both methods on CIFAR-100. The goodness is computed using the equation 2. Our approach is able to preserve the good neighbours while also providing good alternatives to the replaced ones. Consequently, our approach shows higher *goodness* than NNCLR on the non-replaced neighbours, while having a lower score on the replaced ones.

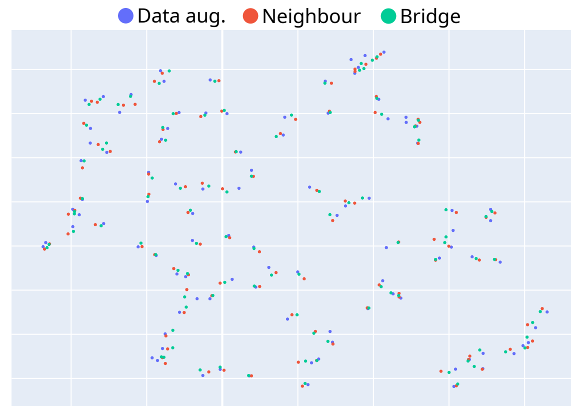

Figure 3. **UMAP visualization** of the best epoch (100 samples).

5.2.3 Qualitative Results

Bridge Point Analysis. We visualize 100 random samples of the best training epoch using UMAP [31] along with their respective neighbours and BP. In Fig. 3, one can see that in several Aug-BP-NN trios, the created BPs are located in the middle of the augmentation and nearest neighbours. This proves the effectiveness of our proposed approach to obtain representations that mostly are representative of both the query augmentation and the extracted neighbour. For the neighbours that are being replaced, new representations are created close to where good neighbours are supposed to be located.

In order to visualize our BPs, we implement a U-Net++ [47] based encoder-decoder architecture for an image reconstruction task. We initialize the encoder part of the U-Net with the weights of our pre-trained encoder and freeze it. Then, we train the decoder for a single epoch. We simulate a previously stored epoch of our pre-trained model by passing the same query and neighbour images through the encoder to obtain their representations and compute the BPs using Equation 4. Once the BPs are computed, we can simply pass them through the decoder to obtain their image visualization. We show the reconstructed queries, neighbours and BPs in Fig. 4. Most of the BPs resemble the original NN, but are enhanced with the characteristics of the query, making them contain information from both NN and queries. The created BPs combine properties such as colours from the query and the neighbour (first row), make mixed samples (second row), or remove portions of the query that are not necessary for the final learning process. Ultimately, we find the resemblance of the examples to the ones that could be obtained by common image augmentation techniques such as MixUp [45]. However, while those techniques augment the images by applying modifications to the pixels, our approach acts directly in the learnt feature space, which is more efficient and completely unsupervised.

Method	CIFAR-10	CIFAR-100	ImageNet-100
NNCLR [14]	92.13	69.19	79.80*
Ours	**92.25**	**70.77**	**80.10**

Table 1. **Linear evaluation results** showing Top-1 Test Accuracy. *- Results extracted from solo-learn [11].

	CIFAR10	CIFAR100
NNCLR	93.11	78.11
Ours	94.76	87.2

Table 2. k**-NN accuracy** showing neighbour retrieval accuracy.

	Replaced NN	Non-replaced NN
NNCLR	-	93.14
Ours	84.66	96.37

Table 3. *Goodness* between queries and extracted neighbours.

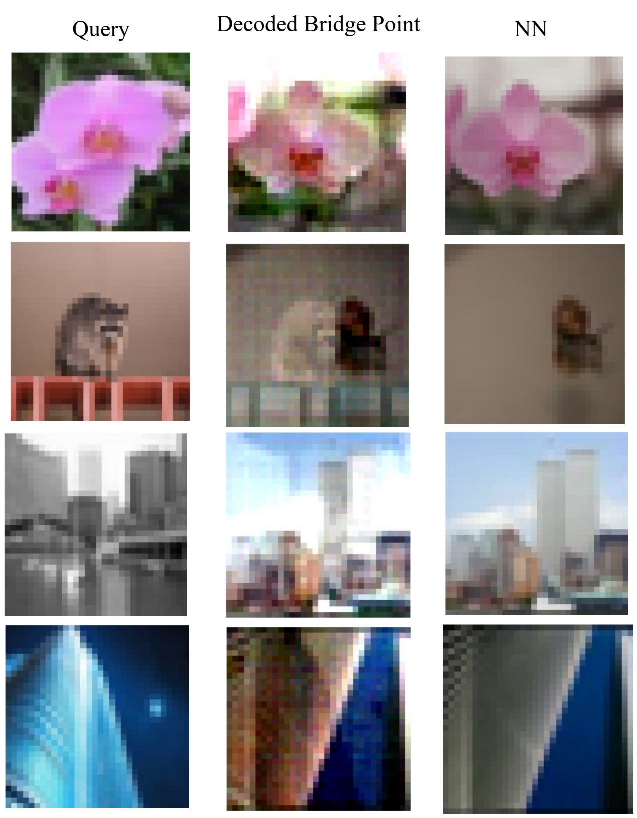

Figure 4. **Decoded BP visualizations** of Query, BP and NN using encoder-decoder image reconstruction.

Neighbour Selection and Replacement. The hypothesis of the existence of bad neighbours consequently implies the existence of neighbours that are good for the CL task and should not be replaced by BPs. We use the histogram of goodness values as in Fig. 5 of all extracted neighbours to analyze the goodness of neighbours. In Fig. 5, we show the histogram for the best epoch of pre-training, which represents the whole training set. As can be seen, while most of the extracted neighbours achieve a very high goodness value, there is still a considerable number of neighbours that possibly do not manage to be good enough for the task. However, deciding an exact threshold that divides good and bad neighbours, is a hard task when we take into account that these values vary during the whole training. At first, when the encoder is not well-trained, the generated representations do not provide the same richness as the ones generated on the final part of the training, making the goodness value fluctuate. For this reason, the selected threshold should be dynamic. In fact, this fluctuation also applies to the different batches that are computed during the training. As can be seen in Figure 6, the mean goodness value (marked in red) deviates depending on the batch. Given these observations, we find that the batch mean threshold is an effective alternative that is dynamic, and adaptative with respect to the training batches.

Finally, we show the effectiveness of the batch-mean threshold using Figure 7. Overall, it can be observed that the replacements align with the notion of a bad neighbour. Most of the bad neighbours belong to a different class while still sharing some features with the original query. However, due to the vast variation of the augmentations used in the pretraining phase, our approach manage to also replace neighbours that could be considered good ones. By looking at their goodness score, we observe that these samples could possibly be very near to the threshold used for that batch. Positively, bridge points tend to share information from both augmentation and the neighbours, therefore the impact of not using the original good neighbour is decreased.

5.3. Ablation Study

We empirically analyze the four components of our approach by a careful ablation study: the representation used as a replacement, the origin of the bridge point used, the replacement strategy type and, finally, the used threshold. For each ablation experiment, we exclusively modify the component to analyse from our best experimental setup. All ablations are done on the CIFAR-100 dataset for a linear classification downstream task.

Replacement Representation Type. In this experiment, we analyse the effect of using different alternatives to the

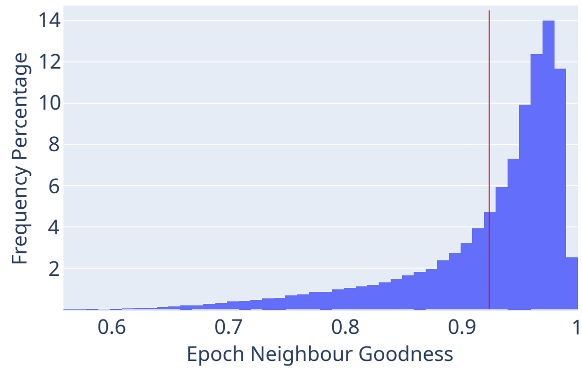

Figure 5. **Histogram of Goodness** for the complete best epoch. The threshold is shown using a red vertical line.

bridge point as a replacement representation. As a first alternative, we replace the bad neighbours with the original query augmentation. As shown in Table 4, this approach manages to outperform the baseline, proving once again the effect of bad neighbours on the model. However, this replacement does not provide any kind of diversity to the contrastive task, and therefore the effect would be the same as switching between NNCLR [14] and SimCLR [8] loss functions depending on the quality of the neighbour. As a second option, we add random Gaussian noise to the query augmentations before contrasting them. This increases the diversity, but it does not produce good evaluation results. The main idea of our proposed bridge points resides in the hypothesis of generating points that could have the potential to be good neighbours i. e. points with proper diversity that would be useful and not confusing to the model. This might not be obtained by the use of simple image augmentations or uncontrolled latent augmentations such as using Gaussian noise.

Bridge Point Type. In this analysis, we explore the effect of using augmentation as the second term in Eq. 4. This way, the bridge point would be based on the query augmentation instead of the neighbour (extrapolation). The bridge point based on the query augmentation does not provide good diversity and, in fact, performs worse than just using the augmentation. This is because the first term is meant to be added to the neighbour for a correct interpolation. Additionally, if we completely interpolate the augmentations instead of just changing the second term, we can observe an improvement. However, it is still less diverse than our proposed bridge point.

Replacement Strategy Type. To prove the effectiveness of our batch mean replacement, we ablate the replacement strategy by experimenting with two different alternatives. First, we show the effects of replacing all neighbours with bridge points. This improves the baseline, however, is held back by the fact that some neighbours do not require a replacement. Good neighbours provide useful information that is even better and more diverse than the bridge point generated. In fact, just randomly replacing half of the neighbours with bridge points is enough to outperform the all-replace alternative. However, a random replacement is less stable compared to the proposed approach.

Method	Top-1
NNCLR	69.19
Replacement Representation Type	
Data Augmentation	69.38
Noisy Data Augmentation	69.10
Bridge Point Type	
Data Augmentation Extrap.	68.85
Data Augmentation Interp.	70.03
Replacement Type	
Replace All Neighbours	69.51
Replace Random Neighbours	70.10
Threshold Type	
Epoch Mean Threshold	69.99
Static Threshold 0.8	69.62
Ours	**70.77**

Table 4. Ablation study.

Threshold Type. We compare our batch mean threshold with a static threshold and a threshold based on the epoch mean. The batch mean threshold provides more dynamism than the epoch mean threshold or the static threshold. For the cases we explored, the additional dynamism of our selected threshold strategy keeps a better balance of the borderline samples than the epoch mean strategy. Depending on the batch, some higher goodness samples are replaced and some lower goodness samples are maintained, which proves to be beneficial for the model. On the contrary, the epoch mean threshold is more restrictive, which leads the model to lower performance. In the case of the static threshold, we do not find it suitable for this task, as it introduces an extra hyperparameter that is very hard to tune in a way that makes the strategy useful for the whole pretraining phase. Overall, our strategy empirically outperforms the other two strategies both in performance and simplicity, as it does not require any further tuning.

5.4. Limitations

While the current study provides valuable insights for NCL, there are still some limitations in the current proposed scheme. We carefully elucidate the limitations that can serve as potential research directions.

Better Threshold Strategy. In the proposed Mending Neighbours approach, we use batch mean threshold due to

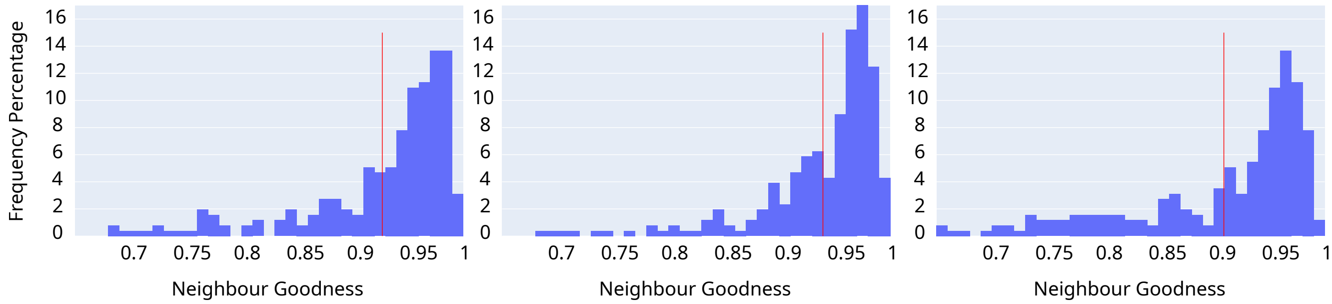

Figure 6. **Histogram of Goodness** for three different batches of the best epoch. The threshold is shown using a red vertical line.

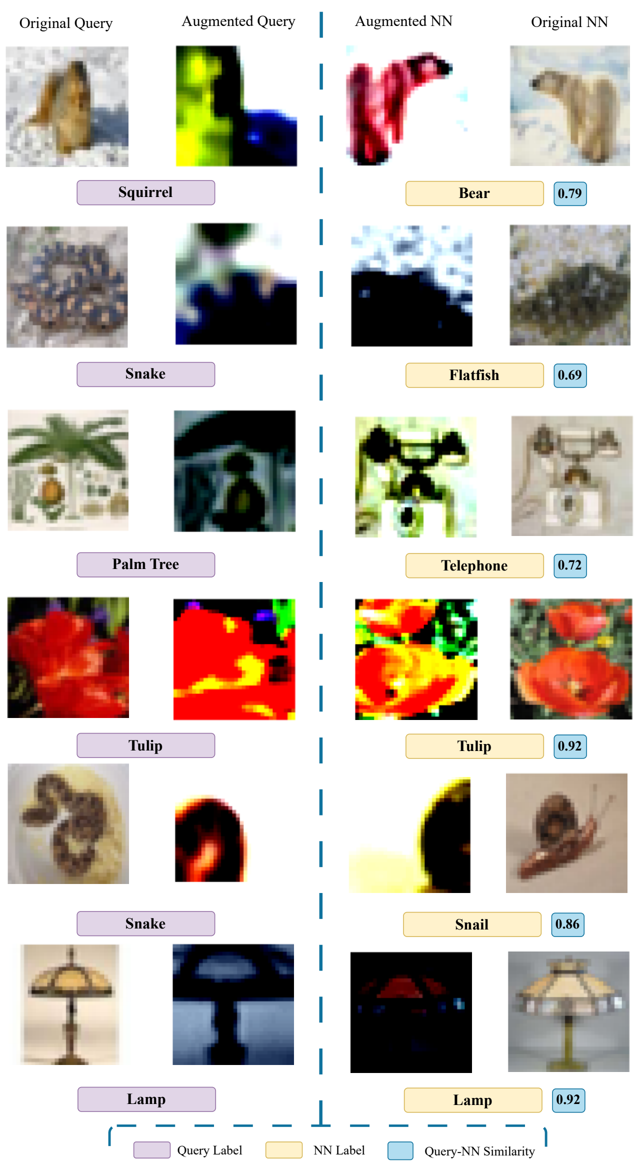

Figure 7. **Replaced NN Visualization.**

its effectiveness and simplicity. However, it is difficult to address borderline neighbours where some good neighbours are also replaced. A better threshold strategy could avoid replacing these neighbours.

Measure of Goodness. Based on our initial hypothesis, we use cosine similarity as a Goodness metric. However, it is simple in terms of providing distance information. A more advanced metric could provide a better measure of goodness and help in better selecting bad neighbours.

Entanglement of Features. We hypothesize that the common entanglement of the features generated by the encoder holds back the whole neighbour selection pipeline. The introduction of a disentanglement process similar to the ones applied in generative algorithms could increase the independence of the features, making them more differentiable and easy to create improved bridge points.

6. Conclusions

In our work, we analyze the current NCL SoTA approaches and identify critical aspects that can affect the performance of NCL algorithms. We propose a novel neighbour correction framework, called "Mending Neighbours" that correctly identifies potential "bad neighbours" and replaces them with a bridge point, a novel representation created directly in the latent space using neighbours and queries. The generated bridge points are more useful than a "bad neighbour" in NCL algorithms and this provides important informative prior information for downstream tasks. We validated our method using different SSL benchmarks and metrics and highlighted our improvements over NNCLR, a popular benchmark NCL algorithm. With in-depth quantitative, qualitative and ablation analysis we showed a measure of neighbour quality and obtained a scheme to identify what constitutes a good neighbour. In future, we plan to generate good neighbours through advanced generative processes that could provide representations of higher quality.

References

[1] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes,

Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023. 1, 2

[2] Ms Aayushi Bansal, Dr Rewa Sharma, and Dr Mamta Kathuria. A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (CSUR)*, 54(10s):1–29, 2022. 1

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[4] Florian Bordes, Randall Balestriero, and Pascal Vincent. Towards democratizing joint-embedding self-supervised learning. *arXiv preprint arXiv:2303.01986*, 2023. 2, 3

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 1, 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 2

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 3, 7

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 1

[10] GE Chongjian, Jiangliu Wang, Zhan Tong, Shoufa Chen, Yibing Song, and Ping Luo. Soft neighbors are positive supporters in contrastive visual representation learning. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3

[11] Victor G Turrisi Da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. Solo-learn: A library of self-supervised methods for visual representation learning. *The Journal of Machine Learning Research*, 23(1):2521–2526, 2022. 4, 5, 6

[12] Terrance DeVries and Graham W. Taylor. Dataset Augmentation in Feature Space. Feb. 2017. 3

[13] Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021. 1

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 2, 3, 4, 5, 6, 7

[15] Imanol G Estepa, Ignacio Sarasúa, Bhalaji Nagarajan, and Petia Radeva. All4one: Symbiotic neighbour contrastive learning via self-attention and redundancy reduction. *arXiv preprint arXiv:2303.09417*, 2023. 2, 3

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 1

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1, 2, 3

[18] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 1

[19] 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*, pages 9729–9738, 2020. 1

[20] Shaoli Huang, Xinchao Wang, and Dacheng Tao. Snapmix: Semantically proportional mixing for augmenting fine-grained data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1628–1636, 2021. 3

[21] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. 1, 2

[22] Amina Kammoun, Rim Slama, Hedi Tabia, Tarek Ouni, and Mohmed Abid. Generative adversarial networks for face generation: A survey. *ACM Computing Surveys*, 55(5):1–37, 2022. 1

[23] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020. 1

[24] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, 2021. 2

[25] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 4

[28] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9090–9098, 2018. 3

[29] 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*, 35(1):857–876, 2021. 1, 2

[30] Zichang Liu, Zhiqiang Tang, Xingjian Shi, Aston Zhang, Mu Li, Anshumali Shrivastava, and Andrew Gordon Wilson. Learning multimodal data augmentation in feature space. *arXiv preprint arXiv:2212.14453*, 2022. 3

[31] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. 5

[32] Juliette Millet, Charlotte Caucheteux, Yves Boubenec, Alexandre Gramfort, Ewan Dunbar, Christophe Pallier, Jean-Remi King, et al. Toward a realistic model of speech processing in the brain with self-supervised learning. *Advances in Neural Information Processing Systems*, 35:33428–33443, 2022. 1

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2

[35] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018. 2

[36] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 1

[37] Ravid Shwartz-Ziv and Yann LeCun. To compress or not to compress–self-supervised learning and information theory: A review. *arXiv preprint arXiv:2304.09355*, 2023. 1

[38] Shashanka Venkataramanan, Ewa Kijak, Laurent Amsaleg, and Yannis Avrithis. Alignmixup: Improving representations by interpolating aligned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19174–19183, 2022. 3

[39] Bram Wallace and Bharath Hariharan. Extending and analyzing self-supervised learning across domains. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 717–734. Springer, 2020. 2

[40] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023. 1

[41] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 1

[42] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 3

[43] Yuhang Zang, Chen Huang, and Chen Change Loy. Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3457–3466, 2021. 3

[44] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 1, 3

[45] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3, 5

[46] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2

[47] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018. 5