Bootstrap Your Own PLM: Boosting Semantic Features of PLMs for Unsuperivsed Contrastive Learning

Yoo Hyun Jeong and **Myeongsoo Han** and **Dong-Kyu Chae**
Department of Artificial Intelligence, Hanyang University, South Korea
{robo0725,myngsoo,dongkyu}@hanyang.ac.kr

Abstract

This paper aims to investigate the possibility of exploiting original semantic features of PLMs (pre-trained language models) during contrastive learning in the context of SRL (sentence representation learning). In the context of feature modification, we identified a method called IFM (implicit feature modification), which reduces the tendency of contrastive models for VRL (visual representation learning) to rely on feature-suppressing shortcut solutions. We observed that IFM did not work well for SRL, which may be due to differences between the nature of VRL and SRL. We propose BYOP, which *boosts* well-represented features, taking the opposite idea of IFM, under the assumption that SimCSE's dropout-noise-based augmentation may be too simple to modify high-level semantic features, and that the features learned by PLMs are semantically meaningful and should be boosted, rather than removed. Extensive experiments lend credence to the logic of BYOP, which considers the nature of SRL. Our code is publicly available at https://github.com/myngsooo/BYOP.

1 Introduction

Contrastive learning has been successfully adopted in the field of VRL by constructing contrastive pairs (drawing positive pairs and repelling negative pairs) based on the sufficient background of augmentation strategies (He et al., 2020; Chen et al., 2020). After that, SRL (sentence representation learning) followed the literature established by the baseline SimCSE (Gao et al., 2021), which proposed to construct contrastive pairs based on *dropout-noise*. Recent studies have generally confirmed the effectiveness of this method (Zhou et al., 2022; Zhang et al., 2022a,b; Wu et al., 2022; Liu et al., 2023).

One interesting point is that SimCSE significantly improves the performance of PLMs (pre-trained language models) on the sentence representation benchmark, named STS benchmark (Cer et al., 2017) where PLMs showed poor performance before the introduction of SimCSE. At the same time, vanilla PLMs have shown comparable or even better performances on several transfer tasks than PLMs trained by SimCSE. We also observed these performance trends, each reported in Table 1 and Table 10 in the Appendix (see the performances of 'Avg.embeddings' and '[CLS] embeddings' which indicate the vanilla PLMs, and that of 'SimCSE').

Based on these empirical results, we hypothesize that PLMs indeed learn several well-represented features, considering their success in the transfer tasks even without the contrastive framework proposed by SimCSE. And such meaningful features would be utilized in contrastive learning of SimCSE, which may partly contribute to the performance improvement in the STS benchmark. Therefore, if there is a way to boost these well-represented features, it would make SimCSE perform even better.

In this context, we identified a method, named IFM (implicit feature modification) (Robinson et al., 2021) from the VRL literature, which tries to *remove* some well-represented features, for the purpose of avoiding *shortcut learning* (Geirhos et al., 2020) – a model tends to depend on a subset of features that is easier to learn during training (Wang and Isola, 2020). We interpret IFM to be the *opposite* of our idea, although IFM ultimately seeks to improve performance as we do. Considering that VRL models are initialized and trained from scratch while PLMs already capture semantic features before contrastive learning, taking a contrary approach to IFM will work better for SRL, rather than following IFM as is.

This study first conducts a pilot study applying the vanilla IFM to SimCSE. Contrary to its success in VRL, we observe a performance degradation, especially for a larger size of PLMs. We interpret that these results come from the fact that PLMs already

Findings of the Association for Computational Linguistics: EACL 2024, pages 560–569

learn several meaningful features, which are indeed helpful in SRL and are not the shortcut features that harm the generalization performance. Then, we propose BYOP (bootstrap[1] your own PLM), which *boosts* the well-represented features, contrary to the intuition of IFM from the VRL perspective. Experimental results demonstrate the effectiveness, robustness, and extensibility of our BYOP.

2 Preliminary

Unsupervised Contrastive Learning for SRL SimCSE followed the literature of the NT-Xent (normalized temperature cross entropy) loss (Chen et al., 2020) with in-batch negatives:

$$l_i = -log \frac{e^{sim(\mathbf{z}_i, \mathbf{z}'_i)/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{z}_i, \mathbf{z}'_j)/\tau}}, \quad (1)$$

where $sim()$, $\mathbf{z}_i$, $\mathbf{z}'_i$, and $\mathbf{z}'_j (i \neq j)$ denotes a similarity function, representation of an anchor instance, a positive pair, and a negative pair. On top of SimCSE, a substantial body of literature has been published that shows promising performance.

Implicit Feature Modification Unlike straightforward supervised learning, the construction of a discriminative instance is an important component in contrastive learning. Contrary to the general belief that lower contrastive loss avoids shortcut solutions (Wang and Isola, 2020), a strong focus on harder instance discrimination can lead to suppression of well-established original features (Robinson et al., 2021). This finding is in line with the reported simplicity bias in supervised learning (Hermann et al., 2020; Huh et al., 2022).

To solve this problem, Robinson et al., 2021 proposed a simple method, called IFM, which accelerates instances to avoid well-represented features by applying adversarial perturbations toward the gradient ascent of the contrastive loss. Considering the similarity function of Equation 1 as a simple ℓ_2-normalized dot product[2], each gradient with respect to the positive ($\nabla_{\mathbf{z}'_i} l_i$) and the negative instance ($\nabla_{\mathbf{z}'_j} l_i$) can be defined as:

$$\nabla_{\mathbf{z}'_i} l_i = \left(\frac{e^{sim(\mathbf{z}_i, \mathbf{z}'_i)/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{z}_i, \mathbf{z}'_j)/\tau}} - 1\right) \cdot \frac{\mathbf{z}_i}{\tau},$$
$$\nabla_{\mathbf{z}'_j} l_i = \frac{e^{sim(\mathbf{z}_i, \mathbf{z}'_j)/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{z}_i, \mathbf{z}'_j)/\tau}} \cdot \frac{\mathbf{z}_i}{\tau}. \quad (2)$$

[1] Same with the popular BYOL (Grill et al., 2020) paper, the term 'bootstrap' is used in its idiomatic sense rather than the statistical sense throughout the paper.

[2] It is an analogous of cosine similarity used in SimCSE.

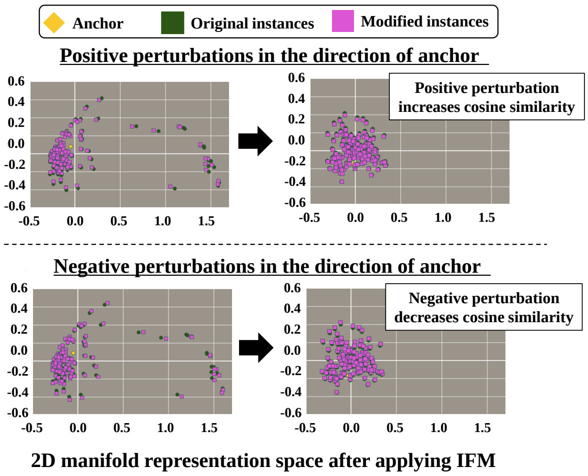


Figure 1: PCA visualization of the 2D representation space using hidden perturbation.

IFM ($l_{i,IFM}$) applies perturbations with a margin (m) toward the direction of gradient ascent ($\nabla_{\mathbf{z}'_i} l_i \propto -\mathbf{z}_i$, $\nabla_{\mathbf{z}'_j} l_i \propto \mathbf{z}_i$) and complements the feature by adopting the multi-task loss $l_{i,total}$. The perturbation loss ($l_{i,IFM}$) and the multi-task loss are computed by:

$$l_{i,IFM} = -log \frac{e^{(sim(\mathbf{z}_i, \mathbf{z}'_i) - m)/\tau}}{e^{(sim(\mathbf{z}_i, \mathbf{z}'_i) - m)/\tau} + \sum_{j \neq i}^{N} e^{(sim(\mathbf{z}_i, \mathbf{z}'_j) + m)/\tau}}),$$
$$l_{i,total} = \frac{1}{2}(l_i + l_{i,IFM}). \quad (3)$$

3 Pilot Study

Despite the effectiveness of IFM in VRL, we assume that boosting the well-represented features, contrary to IFM, will fit in SRL, due to the differences between VRL and SRL; *e.g.*, the use of PLMs that may learn several well-represented features. In this pilot study, we empirically show the failure of the vanilla IFM applied to SimCSE, and provide further analyses to point out differences in the two fields.

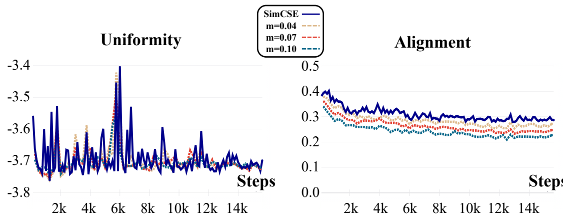
Experimental Setups We followed the settings of SimCSE to tune the basic hyperparameters. For the margin term, we performed a grid search; $m \in [0.01, 0.10]$ with step 0.01. We trained all models for 1 epoch and evaluated them every 250 steps on the STS-B development set to save the best checkpoint. For evaluation, we downloaded the sampled English Wikipedia (10^6) from huggingface (Wolf et al., 2019) same with SimCSE (Gao et al., 2021). We evaluated the following 7 datasets: STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (STS-B) (Cer et al., 2017) and SICK Relatedness (SICK-R) (Marelli et al., 2014).

PLMs	Method	Avg.Score
BERT$_{\text{base}}$	[CLS] embedding	31.40
	Avg. embeddings	52.57
	SimCSE	76.95
	+IFM	77.39
	+BYOPC	77.32
	+BYOPD	**77.45**
	+BYOPC-M	77.32
	+BYOPD-M	77.35
BERT$_{\text{large}}$	[CLS] embedding	32.00
	Avg. embeddings	48.91
	SimCSE	78.46
	+IFM	77.99
	+BYOPC	78.89
	+BYOPD	**79.23**
	+BYOPC-M	79.08
	+BYOPD-M	78.21
RoBERTa$_{\text{base}}$	[CLS] embedding	43.62
	Avg. embeddings	53.49
	SimCSE	76.64
	+IFM	76.97
	+BYOPC	77.62
	+BYOPD	77.43
	+BYOPC-M	77.61
	+BYOPD-M	**77.69**
RoBERTa$_{\text{large}}$	[CLS] embedding	26.64
	Avg. embeddings	52.81
	SimCSE	78.53
	+IFM	77.78
	+BYOPC	78.56
	+BYOPD	78.38
	+BYOPC-M	**78.95**
	+BYOPD-M	78.65

Table 1: Evaluation results of different methods on STS evaluation tasks. Each bold number means the best performance within the PLMs, respectively. ♡ : Results from Gao et al., 2021

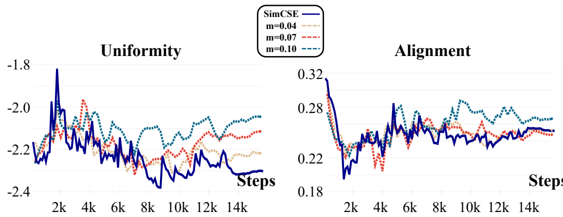
Results and Analyses We report the averaged score of the 7 evaluation tasks performed by SimCSE with the vanilla IFM in Table 1. We observe that IFM improves the performance of SimCSE only in the case of two base models (BERT-base and RoBERTa-base), but shows degraded performance in the two large models. Since the larger size of PLMs have much capacity for establishing useful features during their pre-training, the idea of IFM especially degrades their performances.

Beyond the STS evaluation results, we also investigate the uniformity and alignment metrics (Wang and Isola, 2020) of the STS-B development sets during training, where the former leads to all instances being uniformly distributed and the latter increases the similarity between the anchor and the positive instance. As shown in Figure 3, we can see that the larger margin (m) of IFM leads to larger uniformity and alignment, which generally means degradation. This result is unexpected as there is no meaningful change in uniformity and even there is an improvement in alignment in the training dataset, which we also visualize in Figure 2.



(a) Uniformity (Train) (b) Alignment (Train)

Figure 2: Uniformity and alignment (training) of BERT-base depending on IFM with different margin (m).



(a) Uniformity (b) Alignment

Figure 3: Uniformity and alignment (STS-B) of BERT-base depending on IFM with different margin (m).

Based on the results, we suggest the following intuitions. First, we assume that the dropout-noise-based augmentation is too simple to modify high-level semantic features by IFM. This is a fundamental limitation that makes it difficult to intuitively construct multiple predictive sets of inputs in NLP. In this regard, IFM has difficulty removing frequently used features. Second, as shown in Figure 1, PLMs' semantic spaces are anisotropic – a narrow cone-shaped space (Ethayarajh, 2019; Wang et al., 2019; Li et al., 2020) – before being trained by contrastive learning. We think that IFM's perturbations, positive perturbation (w.r.t. negative instance) and negative perturbation (w.r.t. positive instance) in the direction of the anchor, may be ineffective because PLMs already have some meaningful semantic structures. In other words, PLMs learn some semantic features that are harder to alter by contrastive learning, but still useful for sentence representation.

4 Proposed Method

4.1 BYOP

Motivated by the analyses of the previous section, we propose BYOP (bootstrap your own PLM), which *boosts* semantic features contrary to the concept of IFM. In BYOP, we apply the perturbation in the direction of the gradient *descent*; *i.e.*, additive margin to the positive logits and subtractive margin to the negative logits, opposite to Equation 3.

Perturbation Variants BYOP has two different

PLMs	Method	Avg.Score
$\text{BERT}_{\text{base}}$	SimCSE	75.83 $\pm$ 0.71
	+BYOPD	**76.81** $\pm$ 0.62
	+BYOPD-M	76.43 $\pm$ 0.81
$\text{BERT}_{\text{large}}$	SimCSE	77.14 $\pm$ 1.45
	+BYOPD	**78.98** $\pm$ 0.34
	+BYOPC-M	78.78 $\pm$ 0.30
$\text{RoBERTa}_{\text{base}}$	SimCSE	76.77 $\pm$ 0.06
	+BYOPC	**77.51** $\pm$ 0.21
	+BYOPD-M	77.44 $\pm$ 0.40
$\text{RoBERTa}_{\text{large}}$	SimCSE	78.04 $\pm$ 0.64
	+BYOPC-M	**78.27** $\pm$ 0.65
	+BYOPD-M	78.06 $\pm$ 0.52

Table 2: Averaged results of 3 different random seeds experiments on STS evaluation tasks.

PLMs	Method	Avg.STS
$\text{BERT}_{\text{base}}$	RankCSE-ListMLE	80.11
	+BYOPC	**80.53**
	+BYOPD	80.51
$\text{BERT}_{\text{large}}$	RankCSE-ListMLE	80.24
	+BYOPC	80.64
	+BYOPD	**80.67**
$\text{RoBERTa}_{\text{base}}$	RankCSE-ListMLE	79.05
	+BYOPC	**79.51**
	+BYOPD	79.50
$\text{RoBERTa}_{\text{large}}$	RankCSE-ListMLE	79.70
	+BYOPC	79.53
	+BYOPD	**79.84**

Table 3: Averaged STS results of RankCSE applying BYOP.

types of margin values and 5 candidates for perturbation methods. For the margin value, we use (1) a constant value (BYOPC), which is the same as IFM, and (2) a dynamically changing value (BYOPD), which is determined by the similarity between an anchor and a positive instance. We simply compute the dynamic margin as $\frac{sim(\mathbf{z}_i,\mathbf{z}_i')}{N-1}$ (we set the denominator to $N-1$ to account for the number of in-batch negative samples). For the perturbation method, we explore several combinations of perturbations, which we briefly express as additive '+', subtractive '-', perturbation for positive instance 'p', and perturbation for negative instance 'n'. For example, the additive perturbation for a positive instance and the subtractive perturbation for a negative instance is denoted as 'p+n-' (see Appendix E for their results).

Multi-task Loss VS. Single Loss Following IFM (Robinson et al., 2021), we adopt the multi-task loss (e.g., BYOPD-M) to complement the feature semantics that might be ignored by perturbations. Since BYOP aims to boost the semantic features of contrastive learning, we also conduct experiments for the single loss (i.e., using only the perturbation loss $l_{i,IFM}$). Equation for the two losses is similar to Equation 3 with a subtle change in the margin term. For example, BYOP with 'p+n-' alters each margin term ($+m$ and $-m$) to $sim(\mathbf{z}_i, \mathbf{z}_i') + m$ and $sim(\mathbf{z}_i, \mathbf{z}_j') - m$.

4.2 Empirical Validation

Implementation Details We followed the hyperparameter settings of SimCSE, including batch size, learning rate, and temperature. For BYOP, we performed a grid search to find optimal values such as margin (m) and perturbation types. More detailed settings can be found in Appendix B.

Unsupervised STS Tasks BYOP improves the performance of SimCSE in 4 different PLMs. As shown in Table 1, variants of BYOP lead to better results in most cases: about 0.6% on BERT-base, 1.0% on BERT-large, 1.4% on RoBERTa-base, and 0.5% on RoBERTa-large.

Robustness to Different Seeds Previous work has demonstrated the vulnerability of the unsupervised manner of SimCSE on different random seeds (Jiang et al., 2022). We therefore investigate the robustness of BYOP using multiple random seeds. We first select the best two methods within PLMs based on the results of Table 1, and report the averaged STS results. As shown in Table 2, SimCSE with BYOP shows better performance and also lower standard deviation in most cases.

Applying BYOP to SOTA To assess the extensibility of BYOP, we incorporate BYOP into RankCSE-ListMLE (Liu et al., 2023), a recent state-of-the-art approach in SRL, by using the single loss. As shown in Table 3, it is evident that BYOP plays a significant role in improving performance in all models. These results highlight the potential for BYOP to function as a viable plugin within the contrastive learning schemes.

5 Conclusion

We have proposed BYOP based on the intuition that PLMs' semantic features are useful for sentence representation. Our pilot study, which observes unexpected experimental artifacts in terms of uniformity, also motivates re-examining the logic of the original IFM by boosting the gradient of loss. We have conducted the STS benchmark of which the results back up the assumption of BYOP by testing several variants. We hope that these approaches shed new light on the deeper analysis of the contrastive learning of SRL.

Limitation

Despite its performance, there is a lack of understanding on how the perturbations lead to feature modification in the representation space. The authors of IFM (Robinson et al., 2021) visualized the examples of instances that are the nearest neighbors of modified feature vectors in terms of both positive and negative pairs. In contrast, we do not find any intuitive results in SRL. It seems likely that these results are in fact due to the dropout-based augmentation of SRL, which is much more prone to ignore semantic information when constructing negative pairs.

At present, several research questions remain unclear; which shortcut features of PLMs are harder to remove or can be useful to boost downstream tasks. One of the candidates may be a frequency bias in the representation space (Jiang et al., 2022); *i.e.*, feature vectors align in the space depending on their frequencies. We think that there is ample room for further progress in analyzing these properties, which may lead to the construction of an effective negative pair for SRL.

Due to space limitations, we report results from ablation experiments in the Appendix E. These results include various combinations of perturbations used in BYOP in terms of BYOPD. Similar to SimCSE, we evaluate each method on typical transfer tasks (see Appendix F).

Ethical Consideration

We download all datasets and PLMs used in experiments from huggingface (scholar purpose) to keep intellectual property. Still, ethical issues can be raised such as negative biases which are fundamentally originated from the nature of web-scraped training data (Wiki) (Bender et al., 2021). Furthermore, there are not any other problems which can be critical for the society.

Acknowledgements

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(*MSIT) (No.2018R1A5A7059549) and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01373, Artificial Intelligence Graduate School Program(Hanyang University)). *Ministry of Science and ICT

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *SemEval@ COLING*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

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

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

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. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Katherine Hermann, Ting Chen, and Simon Kornblith. 2020. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33:19000–19015.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. 2022. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. PromptBERT: Improving BERT sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Jiduan Liu, Jiahao Liu, Qifan Wang, Jingang Wang, Wei Wu, Yunsen Xian, Dongyan Zhao, Kai Chen, and Rui Yan. 2023. Rankcse: Unsupervised sentence representations learning via learning to rank. *arXiv preprint arXiv:2305.16726*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. 2021. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2019. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39:165–210.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12052–12066.

Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022a. Unsupervised sentence representation via contrastive learning with mixing negatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11730–11738.

Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. 2022b. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4892–4903.

Kun Zhou, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6120–6130.

	Train	Dev	Test
STS12	-	-	3108
STS13	-	-	1500
STS14	-	-	3750
STS15	-	-	3000
STS16	-	-	1186
STS-B	5749	1500	1379
SICK-R	4500	500	4927

Table 4: Statistics of 7 STS benchmarks from the SentEval toolkit.

A Datasets

Following the literature, we used English Wikipedia, which can be downloaded at Huggingface, and employed the SentEval (Conneau and Kiela, 2018) toolkit for evaluation, where we use 7 STS datasets, which are typical sentence representation benchmarks widely adopted in the SRL field. In addition, we evaluated transfer tasks: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000) and MRPC (Dolan and Brockett, 2005), whose results are reported in Appendix F. Table 4 and Table 5 show the statistics of the datasets.

	Train	Dev	Test
MR	10662	-	-
CR	3775	-	-
SUBJ	10000	-	-
MPQA	10606	-	-
SST-2	67349	872	1821
TREC	5452	-	500
MPRC	4076	-	1725

Table 5: Statistics of 7 transfer task datasets.

B Detailed Implementation

For all cases of BYOP, we perform a grid search to determine the hyperparameters. Specifically, we first define the interval with an extensive search, and then do a grid search within the following range:

- Margin (m) for BYOPC $\in [0.01, 0.1]$, the step size is 0.01.
- Perturbation method $\in$ {p-n-, p+n-, p+, p-, n-}.

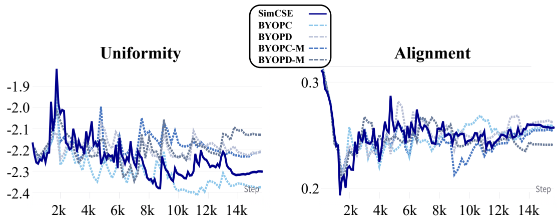
Among combinations of these hyperparameters, we report the settings that show the best performance in STS benchmarks in Table 6. As seen in the table, perturbing the direction of the gradient descent (p+, n-, p-n-, p+n-) shows performance improvement in several cases. Also, applying the perturbations only to positive instances shows performance improvement. We believe this indicates the importance of removing features in positive instances rather than negative instances since in-batch negative samples in unsupervised contrastive learning can lead to the false-negative problem.

C Uniformity and Alignment

Unlike IFM, BYOP aims to boost the gradient of the contrastive loss. In this regard, we first think that the application of BYOP leads to an improvement in uniformity and alignment. However, as shown in Figure 4, where we plot the change of two losses during the training of BERT-base, only BYOPC improves the uniformity and all methods

BYOPC	batch_size	learning_rate	temp (τ)	margin (m)	perturbation
$\text{BERT}_{\text{base}}$	64	3e-5	0.05	0.01	n-
$\text{BERT}_{\text{large}}$	64	1e-5	0.05	0.04	p-n-
$\text{RoBERTa}_{\text{base}}$	128	1e-5	0.05	0.03	p-
$\text{RoBERTa}_{\text{large}}$	256	3e-5	0.05	0.03	p-n-
BYOPD	batch_size	learning_rate	temp (τ)	margin (m)	perturbation
$\text{BERT}_{\text{base}}$	64	3e-5	0.05	–	n-
$\text{BERT}_{\text{large}}$	64	1e-5	0.05	–	p-
$\text{RoBERTa}_{\text{base}}$	128	1e-5	0.05	–	p-
$\text{RoBERTa}_{\text{large}}$	256	3e-5	0.05	–	p-
BYOPC-M	batch_size	learning_rate	temp (τ)	margin (m)	perturbation
$\text{BERT}_{\text{base}}$	64	3e-5	0.05	0.07	n-
$\text{BERT}_{\text{large}}$	64	1e-5	0.05	0.03	p-n-
$\text{RoBERTa}_{\text{base}}$	128	1e-5	0.05	0.005	n-
$\text{RoBERTa}_{\text{large}}$	256	3e-5	0.05	0.02	p+n-
BYOPD-M	batch_size	learning_rate	temp (τ)	margin (m)	perturbation
$\text{BERT}_{\text{base}}$	64	3e-5	0.05	–	p+n-
$\text{BERT}_{\text{large}}$	64	1e-5	0.05	–	p-n-
$\text{RoBERTa}_{\text{base}}$	128	1e-5	0.05	–	p+
$\text{RoBERTa}_{\text{large}}$	256	3e-5	0.05	–	n-

Table 6: Hyperparameters used in the main results (Table 1) of the STS evaluation.



(a) Uniformity (b) Alignment

Figure 4: STS-B development set's uniformity and alignment of BERT-base trained by 4 different BYOP methods.

marginally improve the alignment. This may verify our motivation that the learned shortcut features of PLMs are difficult to remove by the contrastive loss, even in the case of accelerating its gradient.

D Results of STS Benchmark

In this section, we report detailed results of BYOP on the STS benchmark. As shown in Table 7, we can observe that BYOP outperforms the original best result on STS tasks compared to the competing baseline methods based on BERT or RoBERTa. Although BYOP achieves a more visible performance improvement on the base models than on the large models, it still outperforms almost all tasks in both the base and large models. These results suggest that BYOP is effective across different PLMs regardless of their size and different contrastive learning methods.

E Ablational Experiments

We perform additional experiments on the STS evaluation when using different combinations of BYOP. Especially, we report the ablation results of BYOPD, since this method does not require the margin value m. As shown in Table 8 and Table 9, other different methods can also improve the performance of base models, while large models need consideration in the choice of perturbation method since their performance is mostly degraded.

F Results of Transfer Tasks

Following the literature, we also report the performance of 7 transfer tasks as mentioned in Section A. We report these results in Table 10. In general, PLMs show an outstanding performance on downstream tasks despite of their poor capability on STS tasks. In contrast, both SimCSE and BYOP variants show promising performance on STS tasks and also show comparable performance to PLMs. They even outperform in some cases.

PLMs	Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
$\text{BERT}_{\text{base}}$	[CLS] embedding	21.54	32.11	21.28	37.89	44.24	20.29	42.42	31.40
	Avg. embeddings	30.87	59.89	47.73	60.29	63.73	47.29	58.22	52.57
	SimCSE	71.64	82.68	75.81	82.25	78.60	78.93	68.76	76.95
	+BYOPC	71.84	82.86	76.16	82.61	79.07	79.11	69.61	77.32
	+BYOPD	**72.04**	82.86	**76.36**	**82.78**	**79.12**	**79.24**	69.72	**77.45**
	+BYOPC-M	71.67	**82.88**	76.02	82.45	79.09	79.14	**69.98**	77.32
	+BYOPD-M	71.86	82.85	76.23	82.64	79.07	79.13	69.66	77.35
	RankCSE-listMLE	74.53	85.77	78.12	84.71	**81.48**	81.76	**74.37**	80.11
	+BYOPC	76.16	85.97	**78.92**	**84.90**	81.23	82.60	73.91	**80.53**
	+BYOPD	**76.35**	**85.98**	78.82	84.85	81.23	**82.61**	73.71	80.51
$\text{BERT}_{\text{large}}$	[CLS] embedding	27.67	30.76	22.59	29.98	42.74	26.75	43.44	32.00
	Avg. embeddings	27.67	55.79	44.49	51.67	61.88	47.01	53.85	48.91
	SimCSE	70.80	**85.58**	77.34	84.27	79.31	79.07	72.82	78.46
	+BYOPC	**72.45**	85.15	76.42	84.00	**79.56**	80.19	74.43	78.89
	+BYOPD	71.72	85.55	**77.86**	**85.06**	79.08	80.11	75.20	**79.23**
	+BYOPC-M	71.52	84.88	77.37	84.42	79.47	**80.39**	75.50	79.08
	+BYOPD-M	69.80	83.52	76.52	83.61	78.38	79.46	**76.16**	78.21
	RankCSE-listMLE	74.33	86.18	78.75	85.30	**81.07**	81.27	**74.75**	80.24
	+BYOPC	75.59	**86.58**	79.50	**85.74**	80.73	81.86	74.45	80.64
	+BYOPD	**75.61**	86.55	**79.59**	85.71	80.62	**81.99**	74.65	**80.67**
$\text{RoBERTa}_{\text{base}}$	[CLS] embedding	16.67	45.56	30.36	55.08	56.98	38.82	61.90	43.62
	Avg. embeddings	32.11	56.33	45.22	61.34	61.98	55.40	62.03	53.49
	SimCSE	68.65	81.70	73.44	82.30	81.09	80.51	68.76	76.64
	+BYOPC	**70.57**	**82.69**	**74.88**	82.76	**81.66**	**82.04**	68.71	77.62
	+BYOPD	69.92	82.31	74.34	82.29	81.28	81.88	69.99	77.43
	+BYOPC-M	70.44	82.53	74.36	**83.09**	81.65	81.51	69.69	77.61
	+BYOPD-M	70.51	82.49	74.56	82.59	81.61	81.65	**70.44**	**77.69**
	RankCSE-listMLE	**73.45**	84.56	76.00	83.96	**82.67**	82.80	69.89	79.05
	+BYOPC	73.24	84.97	76.79	84.18	82.52	**83.52**	**71.33**	**79.51**
	+BYOPD	73.15	**84.98**	**76.85**	**84.19**	82.49	83.51	71.32	79.50
$\text{RoBERTa}_{\text{large}}$	[CLS] embedding	19.25	22.97	14.93	33.41	38.01	17.30	40.63	26.64
	Avg. embeddings	33.63	57.22	45.67	63.00	61.18	50.59	58.38	52.81
	SimCSE	70.85	83.67	75.83	84.24	80.27	82.42	72.41	78.53
	+BYOPC	70.89	**84.06**	**76.39**	84.52	79.94	82.33	71.77	78.56
	+BYOPD	70.34	83.92	75.50	84.34	80.46	82.17	71.90	78.38
	+BYOPC-M	**72.31**	83.91	76.03	**84.83**	80.12	81.99	**73.43**	**78.95**
	+BYOPD-M	71.79	83.82	76.15	84.36	**80.68**	**82.57**	71.16	78.65
	RankCSE-listMLE	73.69	84.38	76.75	**85.54**	82.18	83.38	72.01	79.70
	+BYOPC	72.84	**84.95**	**77.43**	85.21	80.85	**83.56**	71.84	79.53
	+BYOPD	**74.69**	84.46	76.52	85.36	**82.21**	83.36	**72.31**	**79.84**

Table 7: Results for each method on the STS benchmark. Each bold and underlined number represents the best and second best performance within the PLMs and methods, respectively.

PLMs	Method	Avg.STS	PLMs	Method	Avg.STS
$\text{BERT}_{\text{base}}$	BYOPD	77.45	$\text{BERT}_{\text{large}}$	BYOPD	79.23
	p-n-	77.15		p-n-	77.79
	p+n-	77.11		p+n-	77.36
	p+	77.25		p+	77.80
	p-	75.46		n-	77.76
$\text{RoBERTa}_{\text{base}}$	BYOPD	77.43	$\text{RoBERTa}_{\text{large}}$	BYOPD	78.38
	p-n-	77.10		p-n-	78.20
	p+n-	77.20		p+n-	77.54
	p+	77.24		p+	77.67
	n-	76.56		n-	77.78

Table 8: Ablation results of BYOP equipped with the **single loss**, using different combinations of perturbations on the STS evaluation tasks. The top row within each PLM is the method with the best STS performance, as specified in Table 6.

PLMs	Method	Avg.STS	PLMs	Method	Avg.STS
$\text{BERT}_{\text{base}}$	BYOPD-M	77.35	$\text{BERT}_{\text{large}}$	BYOPD-M	78.21
	p-n-	77.12		p+n-	78.09
	p+	77.03		p+	77.18
	p-	76.80		p-	77.40
	n-	77.29		n-	78.05
$\text{RoBERTa}_{\text{base}}$	BYOPD-M	77.69	$\text{RoBERTa}_{\text{large}}$	BYOPD-M	78.65
	p-n-	77.46		p-n-	77.16
	p+n-	77.09		p+n-	77.36
	p-	77.48		p+	77.85
	n-	76.91		p-	77.49

Table 9: Ablation results of BYOP equipped with the **multi-task loss**, using different combinations of perturbations on the STS evaluation tasks. The top row within each PLM is the method with the best STS performance, as specified in Table 6.

PLMs	Method	MR	CR	SUBJ	MPQA	SST	TREC	MPRC	Avg.
$\text{BERT}_{\text{base}}$	Avg. embeddings	81.50	86.73	95.22	88.02	85.94	90.60	73.68	85.96
	[CLS] embedding	**81.83**	**87.39**	**95.48**	88.21	**86.49**	**91.00**	72.29	**86.10**
	SimCSE	81.37	86.49	94.46	88.66	84.95	87.60	74.32	85.41
	+BYOPC	81.18	86.25	94.49	88.86	84.73	86.80	74.84	85.31
	+BYOPD	81.37	85.94	94.57	88.66	85.01	87.00	**75.01**	85.37
	+BYOPC-M	81.34	86.49	94.63	**89.01**	84.90	86.80	72.75	85.13
	+BYOPD-M	81.17	86.39	94.44	88.79	85.01	86.80	73.16	85.11
$\text{BERT}_{\text{large}}$	Avg. embeddings	84.30	89.22	95.60	86.94	89.29	91.40	71.65	86.91
	[CLS] embedding	**85.89**	**90.15**	**95.83**	86.04	89.95	**93.60**	69.86	87.33
	SimCSE	84.30	87.98	94.86	88.78	89.51	93.00	74.61	87.58
	+BYOPC	84.98	88.08	95.17	89.08	89.73	90.40	75.36	87.54
	+BYOPD	84.53	88.77	95.31	89.26	90.72	92.20	75.01	87.97
	+BYOPC-M	84.80	88.50	95.27	**90.02**	**90.99**	91.40	76.41	88.20
	+BYOPD-M	85.37	88.69	95.13	89.54	**90.99**	92.20	**76.75**	**88.38**
$\text{RoBERTa}_{\text{base}}$	Avg. embeddings	**84.35**	**88.34**	**95.28**	86.13	**89.46**	**93.20**	74.20	**87.28**
	[CLS] embedding	81.27	84.77	94.15	84.18	86.71	81.20	72.17	83.49
	SimCSE	81.75	86.97	93.43	87.28	86.99	84.40	75.01	85.12
	+BYOPC	81.44	86.20	93.03	87.02	86.11	86.20	75.65	85.09
	+BYOPD	82.33	88.08	92.99	87.26	85.89	85.80	**76.12**	85.50
	+BYOPC-M	81.49	87.34	93.25	87.40	87.42	84.60	75.01	85.22
	+BYOPD-M	82.23	87.39	93.41	**87.87**	87.64	85.00	75.42	85.57
$\text{RoBERTa}_{\text{large}}$	Avg. embeddings	**85.46**	**88.85**	**96.04**	88.32	**91.27**	**93.80**	73.74	**88.21**
	[CLS] embedding	83.04	84.58	95.48	86.90	88.47	87.80	69.80	85.15
	SimCSE	83.17	88.40	94.08	**88.57**	87.53	91.20	72.23	86.45
	+BYOPC	81.80	87.42	93.33	88.42	87.20	93.00	**75.77**	86.71
	+BYOPD	82.40	87.18	93.77	88.16	87.10	90.60	74.90	86.30
	+BYOPC-M	80.93	87.47	93.29	88.41	86.00	90.40	75.25	85.96
	+BYOPD-M	82.26	87.26	93.56	88.14	86.44	91.40	74.61	86.24

Table 10: Results of 4 models trained with different methods on transfer tasks. Each bold number and underlined number indicates the best and the second best performance, respectively, within the PLMs. The method named ‘Avg. embeddings’ uses the average of the last layer’s hidden states of PLMs as a sentence representation; the method ‘[CLS] embedding’ uses the last layer [CLS] token’s hidden state of PLMs as a sentence representation.