Pcc-tuning: Breaking the Contrastive Learning Ceiling in Semantic Textual Similarity

Anonymous ACL submission

Abstract

Semantic Textual Similarity (STS) constitutes a critical research direction in computational linguistics and serves as a key indicator of the encoding capabilities of embedding models. Driven by advances in pre-trained language models and contrastive learning techniques, leading sentence representation methods can already achieved average Spearman's correlation scores of approximately 86 across seven STS benchmarks in SentEval. However, further improvements have become increasingly marginal, with no existing method attaining an average score higher than 87 on these tasks. This paper conducts an in-depth analysis of this phenomenon and concludes that the upper limit for Spearman's correlation scores using contrastive learning is 87.5. To transcend this ceiling, we propose an innovative approach termed Pcc-tuning, which employs **P**earson's **c**orrelation **c**oefficient as a loss function to refine model performance beyond contrastive learning. Experimental results demonstrate that Pcc-tuning markedly surpasses previous state-of-the-art strategies, raising the Spearman's correlation score to above 90. [1]

1 Introduction

As a fundamental task within Natural Language Processing (NLP), Semantic Textual Similarity (STS) is not only widely applied across various real-world scenarios including text clustering, information retrieval, and dialogue systems, but also serves as a principal means for evaluating sentence embeddings (Gao et al., 2021).

Sentence embeddings are vector encodings that encapsulate the semantic essence of original texts. Owing to their capacity to facilitate offline computation as well as their pivotal role in realizing retrieval-augmented generation (Zhao et al., 2024), research in this area has garnered considerable attention from numerous institutions and scholars in recent years.

The quality of sentence embeddings is typically assessed via the SentEval (Conneau and Kiela, 2018) toolkit, which measures models based on their average Spearman correlation across seven STS benchmarks. With the continuous advancement of pre-trained language models (PLMs), contrastive learning, and prompt engineering, cutting-edge work in this field has elevated the scores on the leaderboard from an initial 60 (Pennington et al., 2014) to about 86 (Jiang et al., 2023b). As a result, the "PLM + contrastive learning" framework has become the mainstream paradigm in sentence representation research.

However, as illustrated in Table 1, models' performance on standard STS tasks in SentEval appears to have hit a significant bottleneck. Whether utilizing classical discriminative PLMs such as BERT (Devlin et al., 2019) or emerging generative PLMs like LLaMA2 (Touvron et al., 2023b) and Mistral (Jiang et al., 2023a), contemporary state-of-the-art (SOTA) strategies are unable to achieve Spearman's correlation scores higher than 87. Moreover, despite variations in training datasets, contrastive learning loss functions, and model architectures, the final scores are generally similar if the same type of PLM is selected.

In this regard, Li and Li (2023b) posit that PLMs may have reached their performance limits in STS tasks. However, this paper will demonstrate through rigorous mathematical derivation that the core factor causing this performance ceiling is not the inadequacy of PLMs, but the inherent flaws in contrastive learning loss functions. Specifically, contrastive learning only distinguishes between two categories: similar and dissimilar, in determining the semantic relationships between text pairs. This binary classification strategy restricts its maximum achievable Spearman's correlation score to

[1] Our code and checkpoints are available at https://anonymous.4open.science/r/Pcc-tuning.

Methods	PLMs	Spearman
SimCSE	BERT_{110m}	81.57
PromptBERT	BERT_{110m}	81.97
PromCSE	BERT_{110m}	82.13
SuCLSE	BERT_{110m}	82.17
SimCSE ♢	LLaMA2_{7b}	85.24
PromptEOL ♠	LLaMA2_{7b}	85.40
PromCSE ♢	LLaMA2_{7b}	85.70
AngIE ♢	LLaMA2_{7b}	85.96
DeeLM ♢	LLaMA2_{7b}	86.01
PromptEOL ♠	Mistral_{7b}	85.50
PromptSTH ♠	Mistral_{7b}	85.66
PromptSUM ♠	Mistral_{7b}	85.83

Table 1: Average Spearman's correlation scores obtained by SOTA methods on the seven STS benchmarks collected in SentEval. ♢: results from (Li and Li, 2023b). ♠: results from (Zhang et al., 2024).

87.5, even under optimal conditions.

Following this proof, we introduce Pcc-tuning, a novel approach that employs a two-stage training process. This method enhances models' semantic discrimination capabilities by utilizing a small amount of fine-grained annotated data post contrastive learning. With the same 7B-scale generative PLM, Pcc-tuning can achieve an average Spearman's correlation score exceeding 90 on the aforementioned seven STS tasks, significantly surpassing previous best results.

The main contributions of this study are outlined as follows:

- By analyzing the theoretical limits of binary classifiers in STS tasks, we demonstrate that the upper bound of Spearman's correlation scores using contrastive learning methods is 87.5. This finding effectively explains the performance bottlenecks encountered by prior sentence representation strategies.

- Building upon this, we propose Pcc-tuning, a method capable of taking full advantage of fine-grained labeled data with Pearson correlation as its loss function. After fine-tuning PLMs with contrastive learning, we only need to introduce annotated text pairs amounting to 3.7% of the original training set to bring notable performance improvements.

- We extensively validate the effectiveness of Pcc-tuning across internationally recognized STS benchmarks and seven transfer tasks. Experimental results show that Pcc-tuning significantly outperforms previous SOTA methods across different PLMs and prompts.

2 Understanding the Performance Upper Bound of Contrastive Learning

2.1 Contrastive Learning and Binary Classifiers

Currently, leading approaches for sentence representation predominantly center around contrastive learning, with InfoNCE Loss (Oord et al., 2018) being the most commonly adopted loss function. Given an input text x_i, InfoNCE Loss computes the similarity between this sample and its positive example x_i^+ in the numerator, and contrasts it with the similarity calculations between x_i and other texts within the same batch in the denominator. This formulation aims to bring similar instances closer while pushing dissimilar ones apart. The mathematical expression for InfoNCE Loss is presented in Equation 1, where $f(\cdot)$ denotes the encoding method, N represents the batch size, and τ signifies a temperature hyperparameter.

$$\ell_i = -\log \frac{e^{\cos(f(x_i), f(x_i^+))/\tau}}{\sum_{j=1}^{N} e^{\cos(f(x_i), f(x_j^+))/\tau}} \tag{1}$$

Equation 1 reveals that contrastive learning loss functions, exemplified by InfoNCE Loss, essentially classify sentence pairs into two distinct classes: similar and dissimilar. However, no further distinctions are made within these two categories. In other words, as long as x_i is semantically different from x_j or x_k, InfoNCE Loss treats both (x_i, x_j) and (x_i, x_k) as negative sample pairs. As for which of (x_i, x_j) and (x_i, x_k) exhibits a lower degree of similarity, contrastive learning neither concerns itself with this information nor can it readily leverage such details. Indeed, for the majority of embedding models, their training sets are specially adjusted to provide coarse-grained categorical annotations, so as to better align with the contrastive learning framework (Gao et al., 2021).

Therefore, for a set of text pairs $\{(x_i, x_i^?)\}_1^n$, the optimal scenario for contrastive learning methods is to classify the k most similar pairs as positive and the remaining $n-k$ pairs as negative. This setup ensures that there are no inversions in the predicted scores provided by the model. Such an ideal state for contrastive learning models functions similarly to an optimal binary classifier, as illustrated in Figure 1. This classifier segments the dataset into two groups based on a threshold k, assigning a positive label to all samples above the threshold and a negative label to those below. Analyzing the efficacy of this binary classifier reveals the performance boundary of contrastive learning.

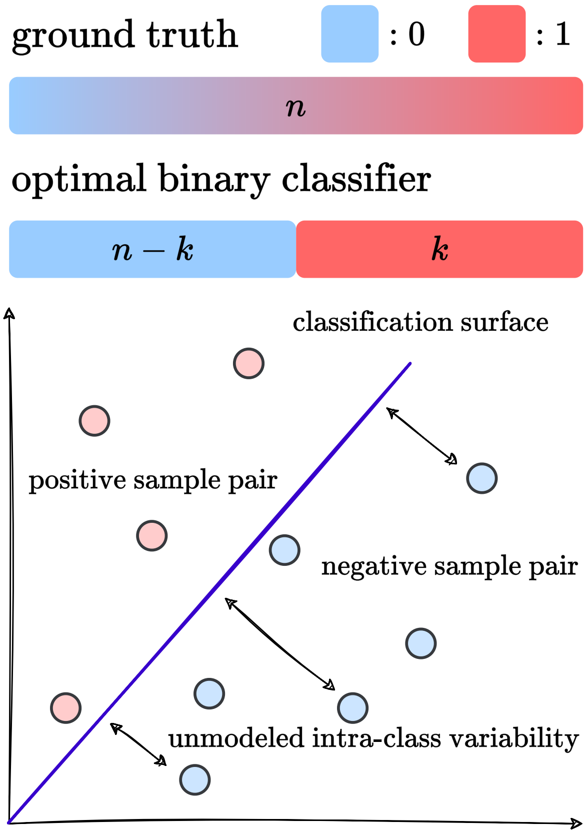

Figure 1: Illustration of the operation of an optimal binary classifier in handling STS tasks. Although the actual similarity scores of the text pairs are a series of floating-point numbers, the binary classifier focuses solely on categorizing them into two classes: similar and dissimilar, without modeling the variability within each category.

2.2 Spearman's Correlation Coefficient

Before deriving the performance upper bound of contrastive learning methods on STS tasks, it is essential to introduce Spearman's correlation coefficient, the primary evaluation metric in this field. This statistic measures the ordinal consistency between the cosine similarity of embeddings and human ratings, as defined by Equation 2:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{2}$$

In this formula, n represents the number of data points, and d_i is the difference between the rank of the i-th sentence pair's cosine similarity after encoding into embeddings and its human-judged similarity rank. Particularly, when multiple entries share the same rating, their ranks are substituted with their mean rank during the computation of Equation 2.

Spearman's correlation coefficient, ranging from $[-1, 1]$, indicates stronger consistency between model outputs and human evaluations as it approaches 1. Typically, the coefficient is multiplied by 100 to yield a percentage score, facilitating more straightforward comparisons of encoding effectiveness across different models.

2.3 The Spearman Correlation Upper Limit of Contrastive Learning Methods

As discussed in Section 2.1, contrastive learning differentiates texts based on binary semantic relations: similar and dissimilar. Thus, its effectiveness parallels that of a binary classifier. This section derives the optimal Spearman correlation achievable by a binary classifier in STS tasks, thereby elucidating the performance upper bound of contrastive learning methods.

Given a collection of text pairs $\mathrm{X} = \{(x_i, x_i^?)\}_1^n$ comprising n samples, we initially arrange the elements of X in descending order according to manually annotated semantic similarity, yielding the sorted set $\mathrm{Y} = \{(y_i, y_i^?)\}_1^n$. Assume that $\cos(y_k, y_k^?) > \cos(y_{k+1}, y_{k+1}^?), \forall k \in [1, n-1]$. Then, for any binary classifier, its performance reaches the optimum only when it categorizes the first k sample pairs of Y as positive examples and the remaining $n-k$ sample pairs as negatives. Otherwise, it indicates at least one misclassification.

Since this binary classifier is solely responsible for constructing an optimal classification boundary between the two categories of similarity and dissimilarity (i.e., distinguishing only whether two texts are semantically akin), its predicted score for the first k samples is consistently identical (assumed to be 1), and likewise for the last $n-k$ samples (assumed to be 0). By the definition of Spearman's correlation coefficient, the difference in rankings between predictions and true values, d_i, alongside $\sum d_i^2$, can be represented as:

$$\begin{aligned} d_i &= i - \frac{k+1}{2}, \quad i = 1, 2, \ldots, k \\ d_i &= i - \frac{k+n+1}{2}, \quad i = k+1, k+2, \ldots, n \end{aligned} \tag{3}$$

$$\sum d_i^2 = \sum_{i=1}^{k}(i-\frac{k+1}{2})^2 + \sum_{i=k+1}^{n}(i-\frac{k+n+1}{2})^2 \tag{4}$$

These equations showcase that $\sum d_i^2$ can be viewed as a function of k. Upon rearranging, we derive: (more details can be found in Appendix C.)

$$\begin{aligned}
\sum d_i^2 &= \sum_{i=1}^{k}(i-\frac{k+1}{2})^2 + \sum_{i=k+1}^{n}(i-\frac{k+n+1}{2})^2 \\
&= \sum_{i=1}^{k}(i-\frac{k+1}{2})^2 + \sum_{i=k+1}^{n}\left((i-\frac{k+1}{2})-\frac{n}{2}\right)^2 \\
&= \sum_{i=1}^{n}(i-\frac{k+1}{2})^2 + (n-k)\frac{n^2}{4} - n\sum_{i=k+1}^{n}(i-\frac{k+1}{2}) \\
&= \sum_{i=1}^{n} i^2 + \frac{n(k+1)^2}{4} - \frac{n(n+1)(k+1)}{2} - \frac{n^2(n-k)}{4} \\
&= \frac{n(n+1)(2n+1)}{6} + \frac{n}{4}\left(k^2 - nk - (n+1)^2\right)
\end{aligned} \tag{5}$$

In Equation 5, n remains constant, thus $\sum d_i^2$ is contingent on $f(k) = k^2 - nk - (n+1)^2$. When $k = \frac{n}{2}$, i.e., when the model deems the first 50% of sample pairs as positives and the latter 50% as negatives, $f(k)$ attains its minimum. Therefore, the minimum value of $\sum d_i^2$ is:

$$\begin{aligned}
\min\left(k^2 - nk - (n+1)^2\right) &= -\frac{5n^2}{4} - 2n - 1 \\
\min(\sum d_i^2) &= \frac{n(n+1)(2n+1)}{6} - \frac{n}{4}(\frac{5n^2}{4} + 2n + 1)
\end{aligned} \tag{6}$$

Subsequently, by substituting $\min(\sum d_i^2)$ into the expression for Spearman's correlation coefficient (Equation 2), the maximum Spearman correlation achievable by this binary classifier is 0.875. This indicates that the optimal performance of contrastive learning in STS tasks will not exceed 0.875.

$$\begin{aligned}
\max(\rho) &= 1 - \frac{n^2-4}{8(n^2-1)} = \frac{7n^2-4}{8(n^2-1)} \\
\lim_{n\to\infty}\max(\rho) &= \lim_{n\to\infty}\frac{7n^2-4}{8(n^2-1)} = \frac{7}{8} = 0.875
\end{aligned} \tag{7}$$

Apart from the original InfoNCE Loss, an extended contrastive learning loss function tailored for NLI datasets (Bowman et al., 2015; Williams et al., 2018), as shown in Formula 8, is frequently utilized in sentence representation research (Gao et al., 2021; Zhang et al., 2023). The incorporation of hard negative example x_j^- in the denominator, equivalent to enlarging the batch size, does not affect the correctness of our derivation.

$$-\log\frac{e^{\cos(f(x_i),f(x_i^+))/\tau}}{\sum_{j=1}^{N}\left(e^{\cos(f(x_i),f(x_j^+))/\tau}+e^{\cos(f(x_i),f(x_j^-))/\tau}\right)} \tag{8}$$

It should be noted that the above conclusion has been validated through numerous experiments. To date, embedding derivation schemes based on contrastive learning have not achieved a Spearman's correlation score above 87. This theoretical analysis provides clear guidance for this empirical observation.

3 Proposed Method

This section introduces Pcc-tuning, an innovative solution for STS tasks. Pcc-tuning employs a two-stage training pipeline and is designed to surpass the 87.5 performance upper bound of contrastive learning methods.

The anisotropy of PLMs' semantic space (Ethayarajh, 2019) presents a longstanding challenge in sentence representation research. Contrastive learning has proven effective in stabilizing embedding distances among semantically similar texts while ensuring a more uniform distribution of vector encodings (Gao et al., 2021), thereby markedly enhancing the semantic space properties of PLMs. Consequently, leveraging contrastive learning to refine the initial state of pre-trained models has emerged as a prevalent strategy within the NLP community (Wang et al., 2022; Li et al., 2023).

Following this established practice, we initially conduct supervised fine-tuning of the PLM using the NLI dataset constructed by SimCSE (Gao et al., 2021). This dataset comprises 275,602 text pairs in triplet form, providing a robust source of coarse-grained labeled information for the model. Our implementation in the first stage closely aligns with that of PromptEOL (Jiang et al., 2023b), where we load the original PLM checkpoint and fine-tune the model with the extended InfoNCE Loss depicted in Equation 8, combined with QLoRA (Dettmers et al., 2024). A unique feature of our methodology is the adoption of the PromptSTH template proposed by Zhang et al. (2024): `"This sentence : '[X]' means something"`, which encapsulates the input sentence [X] and extracts the encoding of the final token as the sentence embedding. Later sections will examine the performance of Pcc-tuning under various prompts.

After the contrastive learning phase, the semantic space of the PLM will be adjusted to a superior

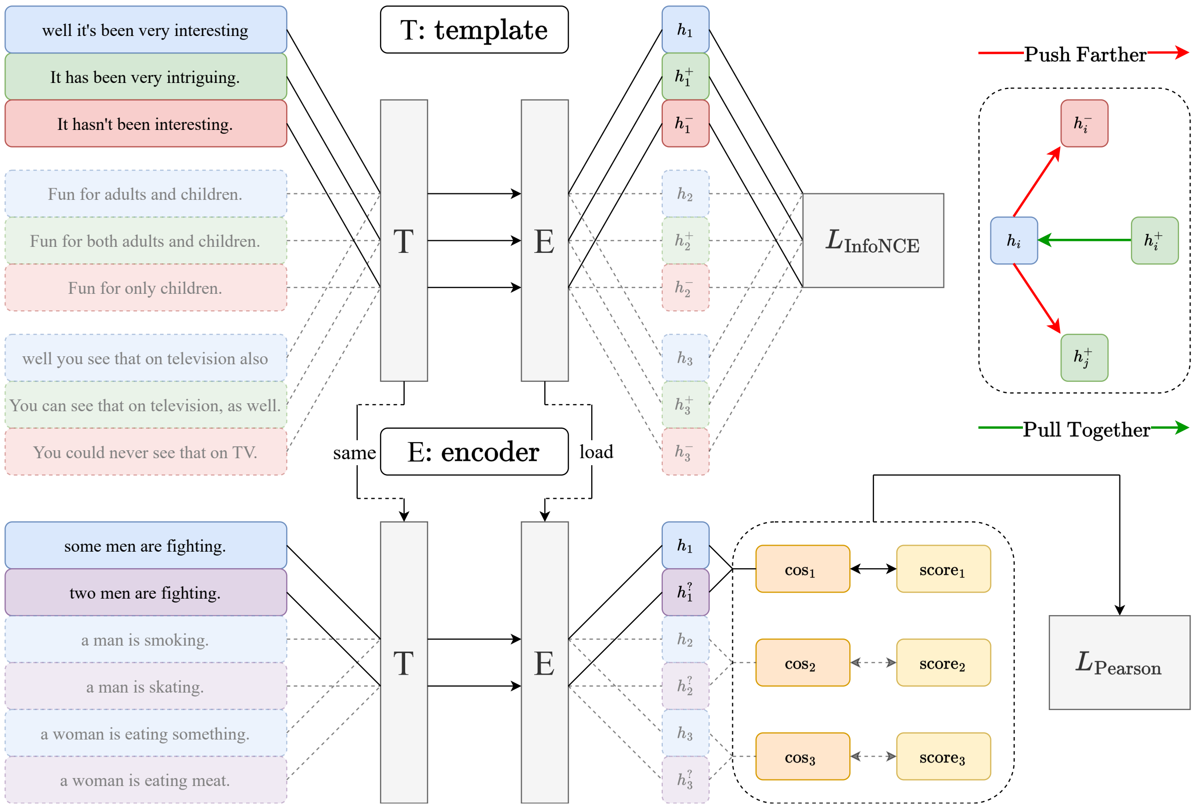

Figure 2: The overall architecture of Pcc-tuning. By default, we use "`This sentence : '[X]' means something`" (Zhang et al., 2024) as the manual template for both stages. In the diagram, h_i denotes the embedding of sentence s_i after model encoding, $\cos_i$ represents the cosine similarity between h_i and $h_i^?$, while score_i is the human-annotated similarity score for s_i and $s_i^?$.

encoding state, capable of generating high-quality embeddings. However, the inability of InfoNCE Loss to harness fine-grained annotation information leads to a pronounced performance bottleneck for contrastive learning methods in STS tasks. To mitigate this issue, a finer distinction is required within the two categories of similarity and dissimilarity, along with introducing the ordinal relationships of text pairs in terms of semantic similarity.

The optimal strategy is to incorporate fine-grained annotated data in the second stage and guide the model's training process via Spearman's correlation coefficient. This ensures maximum consistency between the model's behavior during training and testing phases. However, as Spearman correlation is non-differentiable and thus incompatible with backpropagation, we opt for Pearson's correlation coefficient to update model parameters, which also serves as the inspiration for the name Pcc-tuning. Pearson correlation and our loss function in the second stage are shown in Equation 9, where X represents the cosine similarity between model-derived embeddings, and Y denotes the human-annotated scores for the text pairs.

$$
\begin{aligned}
r &= \frac{\operatorname{cov}(\mathrm{X}, \mathrm{Y})}{\sigma_{\mathrm{X}} \sigma_{\mathrm{Y}}} \\
\ell_p &= -r + 1 \in [0, 2]
\end{aligned}
\tag{9}
$$

Concretely, for a batch of text pairs $\{(x_i, x_i^?)\}_1^N$, we first invoke the PLM to encode x_i and $x_i^?$, obtaining $f(x_i)$ and $f(x_i^?)$. Then, we directly compute their cosine similarity and store the result in $\mathrm{X} = \{\cos(f(x_i), f(x_i^?))\}_1^N$. Subsequently, we input X and the true similarity scores $\mathrm{Y} = \{y_i\}_1^N$ into Equation 9 to calculate the loss.

Employing Pearson coefficient as the loss function enables effective utilization of fine-grained annotation information and supports diverse combinations with a small volume of data. For instance, the tuning dataset in our second stage consists of the training sets of STS-B (Cer et al., 2017) and SICK-R (Marelli et al., 2014), which together contain 10,249 text pairs. This number merely represents 3.7% of the first-stage training dataset, yet their combination varieties reach up to C_{10249}^N. Therefore, even with multiple epochs of training, the similarity ranking of samples in each batch is unlikely to repeat.

Figure 2 provides a detailed illustration of Pcc-tuning's training process. In the first stage, we fine-tune the model using contrastive learning and the NLI dataset. In the second stage, we introduce a small amount of fine-grained annotated data and load the checkpoint from the first phase to further update the model parameters via Pearson's correlation coefficient.

4 Experiments

This section presents the experimental results of Pcc-tuning. Initially, in subsection 4.1, we elaborate on our experimental setup, including evaluation methods, datasets, and the selection of baselines. Subsequently, in subsection 4.2, we compare the performance of Pcc-tuning with contemporary SOTA text representation strategies on internationally recognized STS benchmarks. Finally, in subsection 4.3, we validate the efficacy of Pcc-tuning under diverse prompts.

4.1 Implementation Details

In line with prior studies (Gao et al., 2021; Jiang et al., 2022, 2023b), we utilize the SentEval (Conneau and Kiela, 2018) toolkit to assess our model across seven STS tasks, with Spearman's correlation coefficient as the core metric.

As outlined in Section 3, Pcc-tuning incorporates a two-stage training pipeline. The respective training sets originate from the NLI dataset organized by SimCSE (Gao et al., 2021), containing 275,602 text pairs, and a mixed dataset composed of the training sets from STS-B and SICK-R, totaling 10,249 text pairs. In all experiments, only during the testing phase can models access data from the evaluation benchmarks. It is noteworthy that although Pcc-tuning requires specific corpora at both stages, the total data volume employed is only 285,851 entries. In contrast, the publicly available training data for the current SOTA method, DeeLM (Li and Li, 2023b), includes 480,862 triplet text pairs, with additional data remaining inaccessible.

Our experiments are conducted using several widely adopted 7B-scale generative PLMs: $\text{OPT}_{6.7\text{b}}$ (Zhang et al., 2022), $\text{LLaMA}_{7\text{b}}$ (Touvron et al., 2023a), $\text{LLaMA2}_{7\text{b}}$, and $\text{Mistral}_{7\text{b}}$. To clearly demonstrate the superiority of Pcc-tuning, we primarily compare it against current SOTA strategies. Specifically, among our selected baselines, PromptEOL (Jiang et al., 2023b), PromptSTH (Zhang et al., 2024), AngIE (Li and Li, 2023a), and DeeLM (Li and Li, 2023b) are leading generative PLM sentence representation methods, which significantly outperform BERT-based approaches on STS benchmarks. Meanwhile, openai-ada-002, jina-base-v2 (Günther et al., 2023), and nomic-embed-v1 (Nussbaum et al., 2024) represent the most advanced contrastive learning pre-trained models at present.

4.2 Main Results

Table 2 summarizes the results of the above experiments. Under all tested PLMs, Pcc-tuning consistently transcends the 87.5 Spearman correlation upper bound of contrastive learning methods, achieving an impressive average score of approximately 90. Notably, when employing $\text{Mistral}_{7\text{b}}$ as the backbone, Pcc-tuning attains a Spearman's correlation score of 90.61, substantially surpassing the previous record of 86.01 set by DeeLM. Moreover, Pcc-tuning excels beyond prior SOTA methods in each of the seven STS tasks aggregated within SentEval, manifestly affirming its efficacy. These outcomes collectively underscore the crucial role of modeling fine-grained annotated information in STS tasks.

Furthermore, since Pcc-tuning's first-stage implementation mirrors that of PromptSTH, the comparison between Pcc-tuning and PromptSTH in Table 2 also functions as an ablation study. It reveals that, constrained by the coarse granularity of contrastive learning, whether adopting the earlier released OPT model or the newly open-sourced Mistral model, the Spearman's correlation scores for PromptSTH are confined between 85.3 to 85.7, showing limited progress. In contrast, Pcc-tuning provides improvements of about 5 percentage points, reaffirming the mathematical derivations discussed in Section 2.

In addition to challenges in fully harnessing fine-grained annotated data, another significant drawback of contrastive learning is the need for large batch sizes to prevent model collapse, which consumes substantial computational resources (Jiang et al., 2023b; Zhang et al., 2024). To explore the impact of batch size on Pcc-tuning's performance, we conduct experiments detailed in Appendix A. The findings indicate that Pcc-tuning exhibits strong robustness to varying batch sizes. Additionally, we also assess Pcc-tuning on seven transfer tasks, with outcomes recorded in Appendix B.

Methods	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
Pre-trained Embedding Models								
openai-ada-002 §	69.80	83.27	76.09	86.12	85.96	83.17	80.60	80.72
jina-base-v2 ‡	74.28	84.18	78.81	87.55	85.35	84.85	78.98	82.00
nomic-embed-v1 ‡	65.19	81.67	74.00	83.58	81.87	76.43	75.41	76.88
Fine-tuning Strategies								
Previous SOTA methods. *Implementation on* $\mathrm{LLaMA2}_{\mathrm{7b}}$								
SimCSE ♢	78.39	89.95	84.80	88.50	86.04	87.86	81.11	85.24
PromptEOL	79.24	90.31	84.74	88.72	86.01	87.87	80.94	85.40
AnglE ♢	79.00	90.56	85.79	89.43	87.00	88.97	80.94	85.96
DeeLM ♢	79.01	90.32	85.84	89.47	87.18	89.15	81.08	86.01
Implementation on $\mathrm{OPT}_{\mathrm{6.7b}}$								
PromptSTH	79.30	89.59	84.69	89.17	85.96	88.36	81.51	85.51
Pcc-tuning	**82.83**	**93.30**	**92.66**	**93.09**	**87.44**	**90.34**	**86.24**	**89.41**
Implementation on $\mathrm{LLaMA}_{\mathrm{7b}}$								
PromptSTH	78.48	90.09	85.10	88.71	85.93	88.51	80.95	85.40
Pcc-tuning	**84.40**	**94.40**	**93.15**	**93.49**	**88.62**	**90.86**	**87.08**	**90.29**
Implementation on $\mathrm{LLaMA2}_{\mathrm{7b}}$								
PromptSTH	79.12	89.94	84.54	88.57	86.05	87.82	81.10	85.31
Pcc-tuning	**84.22**	**94.37**	**93.49**	**93.49**	**88.62**	**90.95**	**87.22**	**90.34**
Implementation on $\mathrm{Mistral}_{\mathrm{7b}}$								
PromptSTH	79.19	89.70	85.07	88.88	86.65	88.20	81.95	85.66
Pcc-tuning	**85.77**	**93.79**	**93.78**	**94.02**	**89.07**	**90.73**	**87.14**	**90.61**

Table 2: Spearman's correlation scores across seven STS benchmarks for different methods. This table highlights Pcc-tuning's comprehensive two-stage training strategy in comparison with PromptSTH, which corresponds to the first stage of Pcc-tuning. §: results from (Muennighoff et al., 2022). ‡: results from Zhang and Li (2024). ♢: results from (Li and Li, 2023b).

4.3 Pcc-tuning under Various Prompts

In a pioneering effort to employ generative PLMs for embedding derivation, the Explicit One-word Limitation (EOL) format of the manual template, proposed by PromptEOL (Jiang et al., 2023b), has become the most widely adopted prompt in sentence representation research. Recently, Zhang et al. (2024) introduced two templates that deviate from the EOL structure, namely PromptSTH and PromptSUM. They demonstrated that adherence to the EOL format is not necessary for effective PLM fine-tuning. The specific forms of these prompts are depicted in Table 3, where [X] represents the input text, and the parts highlighted in red signify the positions from which the model extracts embeddings.

To further validate the versatility of our approach, we assess the average Spearman's correlation scores across seven STS tasks using these prompts as the templates for both stages of Pcc-tuning. The corresponding results are delineated in Table 4. It can be seen that regardless of the

PromptEOL
This sentence : "[X]" means in one word:"
PromptSUM
This sentence : "[X]" can be summarized as
PromptSTH
This sentence : "[X]" means something

Table 3: Manual templates employed by PromptEOL, PromptSUM, and PromptSTH. Apart from the differences in prompts, the implementations of these three methods are completely identical.

chosen prompt, Pcc-tuning consistently enhances the model's performance from approximately 85 to around 90, with minimal impact from the different templates on the final outcomes. This finding suggests when applying Pcc-tuning to downstream tasks, there is little need for laborious prompt searches, thereby offering significant application potential.

PLMs	Templates	Stage-1	Stage-2
$\mathrm{OPT}_{6.7b}$	PromptEOL	85.52	89.29
	PromptSUM	85.57	89.39
	PromptSTH	85.51	**89.41**
LLaMA_{7b}	PromptEOL	85.48	**90.38**
	PromptSUM	85.47	90.13
	PromptSTH	85.40	90.29
$\mathrm{LLaMA2}_{7b}$	PromptEOL	85.40	90.32
	PromptSUM	85.53	90.31
	PromptSTH	85.31	**90.34**
$\mathrm{Mistral}_{7b}$	PromptEOL	85.50	90.39
	PromptSUM	85.83	90.57
	PromptSTH	85.66	**90.61**

Table 4: Average Spearman's correlation scores obtained by Pcc-tuning on seven STS benchmarks using different PLMs and manual templates. The settings for stage-1 and stage-2 are consistent with the descriptions in Section 4.1.

5 Related Work

Contrastive learning is currently the principal strategy employed by the NLP community for addressing STS tasks, and our method, Pcc-tuning, is specifically designed to overcome the inherent limitations of contrastive learning, particularly its inability to fully leverage the fine-grained annotated information in text pairs.

Prior to the rise of contrastive learning-based text representation schemes, Sentence-BERT had already proposed enhancing the semantic encoding capabilities of PLMs using the STS-B training set (Reimers and Gurevych, 2019). However, subsequent contrastive learning approaches such as SimCSE (Gao et al., 2021), PromptBERT (Jiang et al., 2022), and CoT-BERT (Zhang et al., 2023) have demonstrated superior performance across the seven STS benchmarks collected in SentEval, thereby making them the focal point of current academic research and development.

Among these efforts, RankCSE (Liu et al., 2023) also recognized that contrastive learning fails to capture the fine-grained ordinal relationships between texts and advocated for the use of Jensen-Shannon divergence to ensure rank consistency of embeddings derived under different dropout masks. However, this technique is only applicable in unsupervised scenarios. Supervised STS solutions, such as PromptEOL (Jiang et al., 2023b), still predominantly employ InfoNCE Loss to update model parameters, thus falling into the performance bottlenecks discussed in this paper.

To the best of our knowledge, this study is the first to propose and substantiate the performance upper bound of contrastive learning methods. Additionally, Pcc-tuning is the inaugural method capable of achieving Spearman's correlation scores above 87 on standard STS tasks, marking a significant advancement in the field.

6 Conclusion

In this paper, we first analyze the structure of contrastive learning loss functions, highlighting that their coarse-grained categorization of semantic relationships between text pairs renders contrastive learning akin to a binary classifier. Building on this insight, we rigorously derive the optimal Spearman correlation achievable by a binary classifier in STS tasks, demonstrating that the upper bound for the Spearman's correlation score of contrastive learning methods is 87.5. This finding effectively explains the performance bottlenecks encountered by current sentence representation methods in STS tasks.

To achieve further breakthroughs, we introduce Pcc-tuning, a strategy that effectively harnesses fine-grained annotated information. Pcc-tuning leverages a two-stage training pipeline and utilizes Pearson's correlation coefficient as the loss function in the second stage to fully exploit the ordinal relationships between text pairs. Extensive experimental results demonstrate that Pcc-tuning significantly enhances the quality of the generated embeddings, and this improvement is consistently observed across different PLMs, prompts, and batch sizes.

Limitations

In preparing the training dataset for the second stage of Pcc-tuning, we employ a mixed corpus composed of the training sets from STS-B and SICK-R. However, the label scales of these two datasets are not completely congruent. Specifically, the STS-B training set contains 5,749 text pairs with similarity scores spanning from 0 to 5, whereas the SICK-R training set includes 4,500 text pairs with similarity scores ranging from 1 to 5. To unify their annotation scales, we transform each label in the SICK-R training set using the formula $5 \times \frac{\text{label}-1}{4}$, thereby converting the labels to the $[0, 5]$ range. Given that this transformation is a simple linear mapping, it is likely that some vital manually annotated information is lost, potentially hindering Pcc-tuning from reaching its optimal performance on the evaluation benchmarks.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

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.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, et al. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023b. Scaling sentence embeddings with large language models. *arXiv preprint arXiv:2307.16645*.

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.

Xianming Li and Jing Li. 2023a. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Xianming Li and Jing Li. 2023b. Deelm: Dependency-enhanced large language model for sentence embeddings. *arXiv preprint arXiv:2311.05296*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13785–13802.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Bowen Zhang, Kehua Chang, and Chunping Li. 2023. Cot-bert: Enhancing unsupervised sentence representation through chain-of-thought. *arXiv preprint arXiv:2309.11143*.

Bowen Zhang, Kehua Chang, and Chunping Li. 2024. Simple techniques for enhancing sentence embeddings in generative language models. *arXiv preprint arXiv:2404.03921*.

Bowen Zhang and Chunping Li. 2024. Advancing semantic textual similarity modeling: A regression framework with translated relu and smooth k2 loss. *arXiv preprint arXiv:2406.05326*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

A Pcc-tuning under Different Batch Sizes

PLMs	Batch Size	Spearman
$\text{OPT}_{6.7\text{b}}$	192	89.34
	216	**89.41**
	224	89.38
	256	89.35
$\text{LLaMA}_{7\text{b}}$	192	90.26
	224	90.28
	232	**90.29**
	256	90.25
$\text{LLaMA2}_{7\text{b}}$	192	**90.34**
	200	90.32
	216	90.30
	256	90.32
$\text{Mistral}_{7\text{b}}$	192	90.54
	224	**90.61**
	232	90.60
	256	90.54

Table 5: Average Spearman's correlation scores achieved by Pcc-tuning on seven STS benchmarks at different batch sizes.

Here, we explore the impact of batch size on the performance of Pcc-tuning. In line with previous sections, we employ four 7B-scale generative PLMs as backbones and report the average Spearman's correlation scores of Pcc-tuning across seven STS tasks in SentEval under various parameter configurations. We continue to utilize PromptSTH as the manual template for encapsulating input sentences, which is also the default setting for Pcc-tuning.

Table 5 presents the results from these experiments. Despite the significant differences between batch sizes of 192 and 256, the resulting Spearman's correlation scores are remarkably similar, with both maintaining high performance levels. This observation indicates that Pcc-tuning is not

Model	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	Avg.
GloVe †	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought ‡	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT †	78.66	86.25	94.37	88.66	84.40	92.80	69.45	84.94
BERT-CLS †	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-BERT ‡	81.09	87.18	94.96	88.75	85.96	88.64	74.24	85.83
SimCSE-BERT ⋆	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
PromptBERT ⋆	80.74	85.49	93.65	89.32	84.95	88.20	76.06	85.49
Implementation on $\mathrm{OPT}_{6.7\mathrm{b}}$								
Pcc-tuning	89.40	92.77	**95.95**	**91.29**	**94.34**	**95.80**	76.00	**90.79**
Implementation on $\mathrm{LLaMA}_{7\mathrm{b}}$								
Pcc-tuning	89.59	92.74	95.63	90.19	94.12	95.00	**77.62**	90.70
Implementation on $\mathrm{LLaMA2}_{7\mathrm{b}}$								
Pcc-tuning	**89.72**	**93.51**	**95.95**	90.75	**94.34**	94.60	76.12	90.71
Implementation on $\mathrm{Mistral}_{7\mathrm{b}}$								
Pcc-tuning	88.78	92.21	95.77	89.45	93.74	**95.80**	74.09	89.98

Table 6: Performance of different methods on seven transfer tasks collected in SentEval. †: results from (Reimers and Gurevych, 2019). ‡: results from (Zhang et al., 2020). ⋆: results from (Jiang et al., 2022).

sensitive to batch size. Further combined with the findings from Section 4.3, where Pcc-tuning exhibits minimal performance fluctuations under different prompts, it can be concluded that Pcc-tuning possesses exceptional robustness and can easily adapt to a variety of hyperparameter configurations.

B Transfer Tasks

In addition to the standard STS benchmarks, we also evaluate Pcc-tuning on several transfer tasks, including MR, CR, SUBJ, MPQA, SST2, TREC, and MRPC. The results, displayed in Table 6, demonstrate that Pcc-tuning consistently outperforms the baselines across all datasets. Notably, its average score exceeds those of SimCSE and PromptBERT by 4 to 5 percentage points, underscoring the ability of Pcc-tuning to generate high-quality embeddings applicable across a broad range of scenarios.

C Derivation Details

Due to space constraints, some steps in the calculation are abbreviated when rearranging Equation 5 in Section 2.3. Here, we provide the complete derivation process:

$$
\begin{aligned}
&\sum d_i^2 \\
&= \sum_{i=1}^{k}(i - \frac{k+1}{2})^2 + \sum_{i=k+1}^{n}(i - \frac{k+n+1}{2})^2 \\
&= \sum_{i=1}^{k}(i - \frac{k+1}{2})^2 + \sum_{i=k+1}^{n}\Big((i - \frac{k+1}{2}) - \frac{n}{2}\Big)^2 \\
&= \sum_{i=1}^{k}(i - \frac{k+1}{2})^2 + \sum_{i=k+1}^{n}\Big((i - \frac{k+1}{2})^2 + \frac{n^2}{4} - n(i - \frac{k+1}{2})\Big) \\
&= \sum_{i=1}^{n}(i - \frac{k+1}{2})^2 + \sum_{i=k+1}^{n}\Big(\frac{n^2}{4} - n(i - \frac{k+1}{2})\Big) \\
&= \sum_{i=1}^{n}(i - \frac{k+1}{2})^2 + (n-k)\frac{n^2}{4} - n\sum_{i=k+1}^{n}(i - \frac{k+1}{2}) \\
&= \sum_{i=1}^{n}(i - \frac{k+1}{2})^2 + (n-k)\frac{n^2}{4} - n\Big(\frac{(n-k)(n+k+1)}{2} - \frac{(n-k)(k+1)}{2}\Big) \\
&= \sum_{i=1}^{n}(i - \frac{k+1}{2})^2 + (n-k)\frac{n^2}{4} - n(n-k)\Big(\frac{(n+k+1)}{2} - \frac{(k+1)}{2}\Big) \\
&= \sum_{i=1}^{n}(i - \frac{k+1}{2})^2 + \frac{n^2(n-k)}{4} - \frac{n^2(n-k)}{2} \\
&= \sum_{i=1}^{n}(i - \frac{k+1}{2})^2 - \frac{n^2(n-k)}{4} \\
&= \sum_{i=1}^{n}\Big(i^2 + \frac{(k+1)^2}{4} - (k+1)i\Big) - \frac{n^2(n-k)}{4} \\
&= \sum_{i=1}^{n}i^2 + \frac{n(k+1)^2}{4} - \frac{n(n+1)(k+1)}{2} - \frac{n^2(n-k)}{4} \\
&= \frac{n(n+1)(2n+1)}{6} + \frac{n}{4}\Big((k+1)^2 - 2(k+1)(n+1) - n(n-k)\Big) \\
&= \frac{n(n+1)(2n+1)}{6} + \frac{n}{4}\Big(k^2 + 2k + 1 - 2(n+1) - 2(n+1)k - n^2 + nk\Big) \\
&= \frac{n(n+1)(2n+1)}{6} + \frac{n}{4}\Big(k^2 - nk - (n+1)^2\Big)
\end{aligned}
\tag{10}
$$