Advancing Semantic Textual Similarity Modeling: A Regression Framework with Translated ReLU and Smooth K2 Loss

Bowen Zhang and Chunping Li
School of Software, Tsinghua University
zbw23@mails.tsinghua.edu.cn, cli@tsinghua.edu.cn

February 9, 2026

Abstract

Since the introduction of BERT and RoBERTa, research on Semantic Textual Similarity (STS) has made groundbreaking progress. Particularly, the adoption of contrastive learning has substantially elevated state-of-the-art performance across various STS benchmarks. However, contrastive learning categorizes text pairs as either semantically similar or dissimilar, failing to leverage fine-grained annotated information and necessitating large batch sizes to prevent model collapse. These constraints pose challenges for researchers engaged in STS tasks that involve nuanced similarity levels or those with limited computational resources, compelling them to explore alternatives like Sentence-BERT. Despite its efficiency, Sentence-BERT tackles STS tasks from a classification perspective, overlooking the progressive nature of semantic relationships, which results in suboptimal performance. To bridge this gap, this paper presents an innovative regression framework and proposes two simple yet effective loss functions: Translated ReLU and Smooth K2 Loss. Experimental results demonstrate that our method achieves convincing performance across seven established STS benchmarks and offers the potential for further optimization of contrastive learning pre-trained models.[1]

1 Introduction

Semantic Textual Similarity (STS) constitutes a fundamental task in natural language processing, wielding significant influence across a multitude of applications, including text clustering, information retrieval, and recommendation systems. Despite the remarkable precision obtained by interactive architectures within these tasks, their inability to support offline computation limits their viability in large-scale text analysis scenarios. In response, the seminal work of Sentence-BERT [**?**] introduces a dual-tower architecture to encode the sentences within a pair separately, thereby facilitating the derivation of independent embeddings. This approach showcases superior efficacy and has rapidly gained widespread acceptance, now serving as a cornerstone for various downstream tasks. Consequently, further improvements to Sentence-BERT hold significant research interest and practical value.

Nevertheless, the advent of contrastive learning methods, exemplified by SimCSE [**?**], has led to more pronounced enhancements on renowned English STS benchmarks, such as STS12-16 [**?**, **?**, **?**, **?**, **?**], STS-B [**?**], and SICK-R [**?**]. This has shifted the research focus in recent years towards integrating contrastive learning techniques with pre-trained language models (PLMs) like BERT [**?**] and RoBERTa [**?**]. An intuitive comparison is that, when employing the NLI dataset [**?**, **?**] as a training corpus, SimCSE-RoBERTa$_{\text{base}}$ attains an average Spearman's correlation score of 82.52 across these STS tasks, hugely surpassing the 74.21 achieved by Sentence-RoBERTa$_{\text{base}}$.

Such discernible performance disparity has inadvertently overshadowed the advantages of Sentence-BERT, especially in terms of data utilization efficiency and computational resource demands. Contrastive learning, by its self-supervised nature, predominantly recognizes text pairs as either similar or

[1] Our code and checkpoints are available at `https://github.com/ZBWpro/STS-Regression`.

dissimilar. This binary categorization restricts contrastive learning methods to training on triplet-form data composed of an anchor sentence, a positive instance, and a hard negative instance in supervised settings [?]. Many practical scenarios, however, tend to provide more finely grained labeled data (e.g., highly relevant, moderately relevant, relevant, and irrelevant) [?], where contrastive learning approaches can usually only exploit text pairs whose similarity indicators are at the endpoints.

Furthermore, since contrastive learning enhances model discriminability by treating other samples within the same batch as negative instances, it requires large batch sizes, thereby consuming substantial computational resources. For example, SimCSE's supervised learning settings include a batch size of 512 and 3 epochs. To accommodate this configuration on consumer-grade GPUs, SimCSE limits the maximum input length to 32 tokens [?]. In contrast, Sentence-BERT and our proposed methodology necessitate a mere batch size of 16 and 1 epoch to reach convergence. Additionally, our default maximum input length is 256, significantly longer than SimCSE's.

The aforementioned drawbacks highlight the difficulty in completely replacing Sentence-BERT with contrastive learning methods. Hence, some cutting-edge works [?] continue to rely on Sentence-BERT for sentence embedding derivation. Nonetheless, given that STS tasks typically categorize text pairs by degrees of semantic similarity, and Sentence-BERT approaches these tasks from a classification standpoint, neglecting the progressive relationships between categories, there exists a clear opportunity for improvement.

As an illustration, consider an STS task with five categories, labeled consecutively from 1 to 5. Traditional classification strategies would yield identical loss for a sample scored at 2, irrespective of its prediction as 3 or 4, an approach evidently suboptimal.

To rectify such deficiency, this paper proposes a novel framework that converts multi-class STS tasks into regression problems, thus effectively capturing the progressive relationships between categories. For a given dataset, we first map its original labels to evenly spaced numerical values, ensuring that samples with higher similarity scores are assigned correspondingly greater values. Then, we set the number of nodes in the output layer to one, thereby enabling the model to produce a continuous prediction. Finally, the model parameters are updated according to the difference between predicted and actual scores.

Distinct from standard regression tasks, the ground truth within our transformed multi-category STS tasks manifest as a series of discrete points along the numerical axis. Therefore, instead of requiring precise matches to the target values, the floating-point predictions just need to be sufficiently close to get correctly classified. To accommodate this characteristic, we introduce a zero-gradient buffer zone to widely utilized L1 Loss and MSE Loss, unveiling two innovative loss functions: Translated ReLU and Smooth K2 Loss.

Comprehensive evaluations across seven STS benchmarks substantiate that our regression framework surpasses traditional classification strategies in handling multi-category STS tasks. Additionally, we find that our approach can further refine the performance of contrastive learning pre-trained models by utilizing filtered STS-B and SICK-R training sets. These findings highlight the effectiveness of our method and underscore the importance of harnessing task-specific data, an aspect often neglected in contrastive learning paradigms.

The main contributions of this study are outlined as follows:

- Building upon the foundation of Sentence-BERT, we develop a regression framework adept at modeling the progressive relationships between categories in multi-class STS tasks. This not only enhances performance but also, due to regression's intrinsic properties, simplifies the prediction process for K-category problems to require only a single output node, significantly minimizing the model's output layer parameter count.

- We propose two novel loss functions, Translated ReLU and Smooth K2 Loss, specifically tailored to address classification problems involving progressive relationships between categories.

- Through empirical evidence, we demonstrate that our strategy can be combined with leading contrastive learning pre-trained models, leveraging fine-grained annotated data to further improve their performance. This offers a new perspective for current research in STS and sentence embeddings.

2 Related Work

In this section, we primarily review three types of STS solutions that are directly relevant to our work:

Siamese Neural Network Architectures: These approaches [?, ?, ?], proposed relatively earlier in the field, have been widely applied across various domains owing to their effectiveness on annotated corpus. Although their performance on the seven STS benchmarks (STS 12-16, STS-B, SICK-R) is generally inferior to contemporary contrastive learning methods, this disparity largely stems from the absence of task-specific training data. Thus, models have the flexibility to opt for alternative sources, such as Wikipedia [?] or NLI datasets [?, ?], which adapt readily to triplet format. Given our goal of tackling multi-category STS tasks, our model architecture remains rooted in the Siamese network. However, in contrast to preceding efforts, we introduce an innovative regression framework specifically designed to capture the progressive relationships between categories.

Contrastive Learning Fine-Tuning Methods: Contrastive learning is currently the mainstream paradigm for addressing STS tasks, with substantial research exploring its integration with the fine-tuning of PLMs [?, ?]. However, contrastive learning loss functions, epitomized by InfoNCE Loss [?], concentrate exclusively on binary semantic categorization and are unable to fully utilize fine-grained labeled texts. Additionally, the necessity for large batch sizes to ensure negative sample diversity and prevent model collapse imposes significant computational demands. These two limitations are inherently difficult to overcome within contrastive learning itself, yet they are precisely the strengths of Sentence-BERT-style dual-tower models. Therefore, a primary objective of this paper is to investigate whether the performance of contrastive learning models can be further enhanced by incorporating traditional Siamese neural network architectures.

Contrastive Learning Pre-Trained Models: With the growing importance of embeddings in retrieval-augmented generation [?] and other application scenarios, more companies and institutions are dedicating efforts to developing specialized text representation models. These approaches generally adopt multi-stage contrastive learning strategies for network pre-training [?, ?, ?]. Additionally, compared to large-scale generative PLMs, lightweight discriminative models that capture bidirectional dependencies are often more preferred. In our experiments section, we employ two state-of-the-art contrastive learning pre-trained models, Jina Embeddings v2 [?] and Nomic Embed [?]. Both are BERT-based encoder architectures with a parameter size of 137 million.

3 Methodology

This section presents our methodological framework, beginning with a detailed exposition of the network architecture and its operational workflow in subsection **??**. Then, in subsections **??** and **??**, we introduce the two novel loss functions proposed in this study.

3.1 Network Architecture

As illustrated in Figure **??**, we utilize a Siamese neural network with shared parameters for encoding input sentences via BERT to obtain corresponding word embedding matrices. Subsequently, sentence embeddings, denoted as $\mathbf{u}$ and $\mathbf{v}$ for paired sentences A and B, are derived through average pooling. These embeddings, both vectors of the hidden dimension, are then concatenated alongside their element-wise difference $|\mathbf{u} - \mathbf{v}|$ and passed through a fully connected layer with parameters sized at $3 \times \text{hidden\_dimension} \times 1$ to produce the model's predicted similarity score.

IMAGE NOT PROVIDED
Caption: Our Regression Framework. Here, the two BERT models share same parameters, with "dim" representing the embedding dimensions of $\mathbf{u}$ *and* $\mathbf{v}$.

Figure 1: Our Regression Framework

Our method diverges from the original dual-tower structures employed by Sentence-BERT and InferSent [**?**] in three critical aspects:

1. We model STS tasks, characterized by a progressive relationship between categories, as regression problems. This is achieved by mapping labels from the original dataset to a sequence of incrementing numbers reflective of their similarity relations, thus conveying to the model that categories are not independent but progressively related.

2. Building on this, we streamline the output node count in the final fully connected layer to one, thereby enabling the model to directly yield a similarity score rather than a categorical probability distribution. Through this adjustment, for STS tasks with K categories, we effectively reduce the parameter size of the output layer from $3 \times \text{hidden\_dimension} \times K$ to $3 \times \text{hidden\_dimension} \times 1$. In light of the expanding hidden layer dimensions in modern PLMs, this optimization can save considerable computational resources.

3. Unlike the classification-based approach of InferSent and Sentence-BERT, which assigns target classes for sentence pairs according to the highest probability, our regression framework categorizes based on the closeness between the predicted and actual values.

To better understand this process, consider an STS task with four categories: "highly relevant," "moderately relevant," "slightly relevant," and "irrelevant." After clarifying the progressive relationship between these categories, we would map them to four consecutive numbers 0, 1, 2, 3, respectively, ranging from "irrelevant" to "highly relevant." This mapping strategy is flexible, allowing for task-specific adjustments in both numerical nodes and intervals. Furthermore, the mapped nodes do not necessarily have to be integers. Subsequently, we encode the paired sentences and compute their semantic similarity, resulting in a floating-point prediction. By rounding this value, it can be converted into the corresponding label. For instance, a prediction of 2.875 for a sample pair would be classified as "highly relevant," as it is closest to the boundary point of 3. Similarly, if a sample receives a predicted value of 1.333, it would be approximated to 1 and thus classified as "slightly relevant," because 1.333 is closer to 1 among the four boundary points 0, 1, 2, 3.

Extending from the above examples, it can be seen that if the original labels are mapped to nodes spaced by d, as long as the difference between the model's prediction and the ground truth is less than $\frac{d}{2}$, the sample will be correctly classified. Specifically, for consecutive natural numbers, d is equal to 1. However, conventional regression loss functions, represented by L1 Loss and MSE Loss, always enforce the model to exactly match the true value, a requirement that is unnecessary for our task scenario. Thus, we introduce a zero-gradient buffer zone into both functions, unveiling two new loss functions: Translated ReLU and Smooth K2 Loss.

3.2 Translated ReLU

We first present Translated ReLU, mathematically formulated in Equation **??**. Herein, d represents the interval between mapped category labels. As previously discussed, when the difference between the model's predicted value and the ground truth is less than $\frac{d}{2}$, it signifies a correct classification of the sample. Traditional regression loss functions, however, mandate absolute congruence between predictions and true values, applying a penalty for any deviation. This stringent requirement to some extent diverts the model's focus from difficult samples that have not yet been correctly classified and ignores the inherent variability within classes.

$$
\begin{aligned}
x &\to |\text{prediction} - \text{label}| \geq 0 \\
f(x) &= \begin{cases} 0 & x < x_0 \leq \frac{d}{2} \\ k(x - x_0) & x_0 \leq x \end{cases} \\
f(x) &= \max(0, k(x - x_0))
\end{aligned}
\tag{1}
$$

To circumvent this limitation, we introduce an adjustable threshold hyperparameter x_0, and set the loss function to zero for values within $[0, x_0]$. This modification posits that a divergence less than x_0 between prediction and ground truth is deemed sufficiently precise, thus exempt from penalty or gradient update. For disparities exceeding x_0, Translated ReLU imposes a linear penalty. To maintain accurate classification, x_0 must not exceed $\frac{d}{2}$, with the interval between x_0 and $\frac{d}{2}$ acting as a margin akin to that in Hinge Loss. This margin can enhance model robustness by penalizing correctly predicted samples that lack adequate confidence. Additionally, a parameter k is specified to control the slope of the function.

The graphical depiction of Translated ReLU is exhibited on the left side of Figure **??**, with parameters set to $k = 2$ and $x_0 = 0.25$. This configuration resembles the ReLU activation function, albeit with a rightward translation. Our study employs Translated ReLU as a loss function and will compare its effects with those of L1 Loss in ensuing sections to demonstrate the significance of zero-gradient buffer zone for augmenting model performance.

3.3 Smooth K2 Loss

Translated ReLU is characterized by its simplicity and efficacy. Nonetheless, we acknowledge its limitation pertaining to the abrupt lack of smoothness at the demarcation point $x = x_0$, alongside a constant gradient that fails to accommodate varying strengths of updates based on the distance between predictions and actual values. To address these concerns, we introduce another loss function termed Smooth K2 Loss to provide a smoother transition and a gradient that dynamically adjusts in accordance with the magnitude of discrepancy from the ground truth. The formulation and the derivative of Smooth K2 Loss are specified as follows:

$$
\begin{aligned}
x &\to |\text{prediction} - \text{label}| \geq 0 \\
f(x) &= \begin{cases} 0 & x < x_0 \leq \frac{d}{2} \\ k(x^2 - 2x_0x + x_0^2) & x_0 \leq x \end{cases} \\
\frac{\partial f(x)}{\partial x} &= \begin{cases} 0 & x < x_0 \leq \frac{d}{2} \\ 2k(x - x_0) & x_0 \leq x \end{cases}
\end{aligned}
\tag{2}
$$

Echoing the design of Translated ReLU, Smooth K2 Loss also incorporates a zero-gradient buffer zone, but exhibits a quadratic function for $x \geq x_0$, as illustrated on the right side of Figure **??**. Given the differential mathematical underpinnings of these two loss functions, Smooth K2 Loss is recommended for scenarios with high-quality data and strong credibility. In contrast, when dealing with datasets that contain considerable noise, Translated ReLU may be a more suitable choice.

Additionally, prior to the application of Translated ReLU and Smooth K2 Loss, it is advisable to consider reassigning prediction values that transcend the defined category range to the nearest boundary. For instance, in a classification task where the category labels can be sequentially converted to 0, 1, 2 and 3, if the model predicts a value of 3.57 for a sample with an actual label of 3, this might be deemed acceptable and potentially obviate the need for a loss adjustment. This rationale stems from the observation that, despite the prediction's deviation exceeding $\frac{d}{2} = 0.5$, the absence of subsequent boundary points beyond 3 warrants a relaxation of this criterion.

IMAGE NOT PROVIDED
Caption: Comparison of Translated ReLU and Smooth K2 Loss, both with $k = 2$, $x_0 = 0.25$.

Figure 2: Comparison of loss functions

4 Experiment

This section provides empirical validation of our regression framework and two innovative loss functions. We commence by comparing the performance of different modeling strategies for multi-category STS tasks and various loss functions (subsection **??**). Next, we demonstrate that, when supplemented with fine-grained training data, our Siamese neural network can effectively enhance the performance of contrastive learning PLMs (subsection **??**). Following this, we highlight the computational efficiency of our methodology (subsection **??**) and explore the influence of varying hyperparameter settings on model performance (subsection **??**). Finally, subsection **??** presents ablation studies on our network architecture.

4.1 STS Performance Based on Traditional Discriminative Pre-Trained Models

Our experimental setup here closely mirrors that of Sentence-BERT, leveraging fine-tuning on BERT or RoBERTa with a composite corpus derived from the SNLI and MNLI datasets. These NLI datasets categorize sentence pairs into three distinct classes: contradiction, neutral, and entailment. Sentence-BERT maps these classes to 0, 2, and 1, respectively, and employs a classification strategy for training [**?**]. In contrast, our method sequentially maps contradiction, neutral, and entailment to 0, 1 and 2. This mapping reflects the natural order of semantic similarity, from least to most similar, thereby enabling our regression framework to better capture the progressive relationships between categories.

For computational efficiency, we uniformly set the batch size to 16 and limit training to a single epoch, with model checkpoints saved based on performance metrics on the STS-B development set. The specific hyperparameter settings for Translated ReLU and Smooth K2 Loss are detailed in Table **??**. During evaluation, we assess the model's average Spearman correlation across seven STS tasks via the SentEval toolkit [**?**].

Table 1: Hyperparameter configurations for our two loss functions when fine-tuning BERT and RoBERTa on the NLI dataset.

PLM x_0	Loss	k
BERT$_{\text{base}}$ 0.25	Translated ReLU	2.5
BERT$_{\text{base}}$ 0.25	Smooth K2 Loss	2
RoBERTa$_{\text{base}}$ 0.25	Translated ReLU	1
RoBERTa$_{\text{base}}$ 0.25	Smooth K2 Loss	3

The results of these experiments are summarized in Table **??**, from which we distill insights along three pivotal aspects:

Table 2: Spearman's correlation scores for different methods across seven STS tasks. This table is partitioned to facilitate a single variable comparison. ♣: results from [?].

Models	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
Implementation on BERT$_{\text{base}}$								
Sentence-BERT$_{\text{base}}$ ♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
BERT$_{\text{base}}$ + Cross Entropy	70.01	71.18	70.10	78.37	72.92	74.88	73.58	73.01
BERT$_{\text{base}}$ + L1 Loss	69.76	69.56	68.13	76.33	70.96	73.61	70.28	71.23
BERT$_{\text{base}}$ + Translated ReLU	72.51	75.46	72.34	78.46	72.64	76.54	72.02	74.28
BERT$_{\text{base}}$ + MSE Loss	72.38	76.47	74.35	78.71	72.95	77.91	70.67	74.78
BERT$_{\text{base}}$ + Smooth K2 Loss	72.39	78.33	75.28	80.26	74.52	78.78	72.65	76.03
Implementation on RoBERTa$_{\text{base}}$								
Sentence-RoBERTa$_{\text{base}}$ ♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
RoBERTa$_{\text{base}}$ + Cross Entropy	71.15	74.29	72.66	79.44	74.12	76.56	73.02	74.46
RoBERTa$_{\text{base}}$ + L1 Loss	68.12	62.27	64.20	72.80	67.28	72.44	66.82	67.70
RoBERTa$_{\text{base}}$ + Translated ReLU	71.13	76.07	72.18	78.13	73.94	77.59	70.94	74.28
RoBERTa$_{\text{base}}$ + MSE Loss	72.67	77.09	72.93	79.52	74.12	77.88	69.85	74.87
RoBERTa$_{\text{base}}$ + Smooth K2 Loss	72.53	78.28	73.88	80.88	75.35	77.44	73.94	76.04

1. **Classification Strategy vs. Regression Strategy:** Our regression framework, particularly when utilizing Smooth K2 Loss, yields an average Spearman correlation of 76.03 for BERT$_{\text{base}}$ and 76.04 for RoBERTa$_{\text{base}}$. These figures significantly outstrip those attained through Sentence-BERT and the classification strategy with Cross-Entropy Loss, highlighting the regression-based modeling's superiority in both reducing the output layer's parameter size and enhancing semantic discrimination in multi-category STS tasks.

2. **Efficacy of the Zero-Gradient Buffer Zone:** The adoption of Translated ReLU improves performance for both BERT and RoBERTa beyond what is achieved with L1 Loss. Similarly, employing Smooth K2 Loss surpasses MSE Loss on both PLMs. These comparisons underline the benefit of incorporating a zero-gradient buffer zone, which helps balance the model's attention across diverse samples in regression-modeled multi-category classification tasks.

3. **Adaptive Gradients Aligned with Prediction Errors:** Models trained with Smooth K2 Loss outperform those utilizing Translated ReLU, and models employing MSE Loss exceed those with L1 Loss. This evidences the advantages of dispensing differentiated gradients in line with prediction-ground truth deviations, especially when leveraging high-quality datasets like NLI.

Collectively, these findings substantiate the merit of (1) adopting a regression framework for multi-class STS tasks and (2) enhancing traditional regression loss functions with a zero-gradient buffer zone to optimize model performance.

4.2 STS Performance Based on Contrastive Learning Pre-Trained Models

While the Siamese neural network, augmented by our regression framework and innovative loss functions, has exhibited significant performance improvements, a gap remains when compared to leading contrastive learning methods. To address this, we exploit the strengths of Siamese architectures in fully utilizing annotated data and explore whether it can be combined with top-performing contrastive learning models.

Jina Embeddings v2 [?] and Nomic Embed [?] are two recently released embedding models that employ multi-stage contrastive learning strategies during pre-training, combining supervised and unsupervised approaches to optimize the networks. Both have achieved state-of-the-art results on the MTEB leaderboard [**?**]. Therefore, if our method can further enhance the performance of these models, it would provide valuable insights for future research.

Among the seven STS benchmarks (STS12-16, STS-B, and SICK-R), STS-B and SICK-R come with their own training datasets. Specifically, STS-B contains sentence pairs with similarity scores ranging from 0 to 5, while SICK-R includes pairs with scores from 1 to 5. To ensure accurate evaluation, we performed strict data filtering to remove any training text pairs that appeared in the test sets. Details of this filtering process are provided in Appendix **??**. We then applied a linear transformation, $\frac{5\times\text{label}(z)-1}{4}$, to convert all SICK-R training labels to the range $[0, 5]$ and merged them with the filtered STS-B training set. This procedure resulted in a fine-grained, task-specific corpus containing 5,398 sentence pairs.

Since Jina Embeddings v2 and Nomic Embed have undergone pre-training on massive texts, their model parameters have favorable initial distributions. In contrast, our newly introduced linear layer is randomly initialized (Figure **??**). To facilitate effective joint training, we first freeze the entire PLM and only update the linear layer using the NLI dataset described in section **??**. After completing this step, we optimize both the PLM and the linear layer with the filtered STS training data. A schematic diagram of this workflow is shown in Figure **??**. Throughout the entire procedure, Smooth K2 Loss is employed as the loss function.

IMAGE NOT PROVIDED

Caption: Our two-stage fine-tuning process for contrastive learning pre-trained models. In the figure, modules highlighted in red are active during training and undergo backpropagation, while modules in blue are frozen and do not carry out updates.

Figure 3: Two-stage fine-tuning process

The performance of Nomic Embed and Jina Embeddings v2 on the seven STS tasks before and after fine-tuning is presented in Table **??**. The results demonstrate that our network framework effectively enhances the performance of both models and surpasses BERT-based methods with comparable parameter sizes. Notably, we also test the impact of further updating the PLM using contrastive learning, which requires additional processing of the 5,398 training samples obtained earlier. To illustrate this, we take InfoNCE Loss [**?**], the most widely adopted contrastive learning loss function, as an example.

For any input sentence x_i, InfoNCE Loss computes the similarity between its encoding $f(x_i)$ and that of its positive instance $f(x_i^+)$ in the numerator, while aggregating the similarity calculations between $f(x_i)$ and other samples within the same batch in the denominator. This formulation aims to bring similar samples closer and push dissimilar ones apart. Equation **??** presents the standard expression of InfoNCE Loss, where N represents the batch size and τ denotes a temperature hyperparameter.

$$\ell_i = -\log \frac{e^{\cos(f(x_i), f(x_i^+))/\tau}}{\sum_{j=1}^{N} e^{\cos(f(x_i), f(x_j^+))/\tau}} \tag{3}$$

As indicated by Equation **??**, the only component of InfoNCE Loss that can be filled with labeled data is the similarity calculation between positive samples in the numerator. Consequently, contrastive learning is limited to utilizing only text pairs with the highest similarity ratings. To work within this constraint, we selected 1,543 samples from the 5,398 training pairs by adopting a threshold of 4.0 to filter out positive sample pairs. As it can be observed in Table **??**, after discarding such a large portion of annotation information, contrastive learning yields little improvement and may even lead to model collapse, causing performance degradation. In contexts where more detailed, domain-specific data is available, the shortcomings of contrastive learning in not being able to effectively harness multi-level label information, only performing coarse semantic distinctions, becomes more pronounced.

Table 3: Spearman's correlation coefficients of different methods across seven STS tasks. The "+Contrast" notation in the first column refers models further fine-tuned with contrastive learning. ♠: results from [**?**]. ♡: results from [**?**]. ♢: results from [**?**].

Models	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
CT-SBERT$_{\text{base}}$ ♠	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
SimCSE-BERT$_{\text{base}}$ ♠	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
PromptBERT$_{\text{base}}$ ♡	75.48	85.59	80.57	85.99	81.08	84.56	80.52	81.97
PromCSE-BERT$_{\text{base}}$ ♢	75.58	84.33	79.67	85.79	81.24	84.25	80.79	81.81
Nomic Embed Text v1	73.75	85.03	80.52	87.40	83.55	83.90	76.52	81.52
Nomic Embed Text v1 + Contrast	76.10	85.79	80.58	87.35	83.54	85.16	72.33	81.55
Nomic Embed Text v1 + Ours	73.06	86.63	81.06	87.67	83.43	85.18	82.75	82.83
Jina Embeddings v2	74.28	84.18	78.81	87.55	85.35	84.85	78.98	82.00
Jina Embeddings v2 + Contrast	76.04	86.37	80.16	86.53	85.24	84.31	74.18	81.83
Jina Embeddings v2 + Ours	75.17	86.10	79.96	88.44	85.01	86.83	83.34	83.55

4.3 Computational Resource Overhead

In addition to its inability to fully leverage fine-grained annotated data, the high memory requirements of contrastive learning also pose a challenge for many researchers. In this section, we compare the computational resource consumption of our method with that of SimCSE during training, based on four 24GB NVIDIA GPUs. The results are summarized in Table **??**, where both BERT and RoBERTa are the base versions.

Despite setting the maximum sequence length for SimCSE to approximately 40% of our method's default configuration, its memory usage remains significantly higher, reaching an astonishing 81GB. Thus, overall, our Siamese neural network strategy is more suitable for resource-constrained environments.

Table 4: Computational demands of our method compared to SimCSE during the training phase. The third column, "Length," represents the maximum sequence length supported by each model (cutoff length).

PLMs	Method	Length	Memory
BERT	SimCSE	100	81.30 GB
	Ours	256	41.27 GB
RoBERTa	SimCSE	100	81.61 GB
	Ours	256	42.33 GB

4.4 Impact of Different Hyperparameter Settings

In this study, we introduce two novel loss functions, Translated ReLU and Smooth K2 Loss, each characterized by two critical hyperparameters: k and x_0. The parameter k primarily controls the gradient of the loss function, while x_0 defines the tolerance threshold for model predictions. To discern the influence of these hyperparameters on model performance, we conducted a series of experiments across both traditional discriminative PLMs (BERT, RoBERTa) and the latest contrastive learning PLMs (Nomic Embed v1, Jina Embeddings v2).

The outcomes of these investigations are consolidated in Table **??**. Rather than executing an exhaustive grid search, initial values were selected based on our preliminary insights, followed by incremental adjustments. This implies that there may still be room for further improvement in our model's performance.

The experimental results from Table **??** reveal minor fluctuations in model performance across diverse hyperparameter configurations, which affirms the resilience and robustness of our proposed methodology. This stability highlights the inherent adaptability of our regression framework as well as loss functions, suggesting their applicability to a wide range of modeling scenarios without necessitating extensive hyperparameter optimization.

Table 5: Average Spearman's correlation scores across seven STS tasks under different values of k and x_0.

PLM	Loss	k	x_0	Performance
Implementation on Traditional Discriminative PLMs				
BERT$_{\text{base}}$	Translated ReLU	1.5	0.25	74.21
BERT$_{\text{base}}$	Translated ReLU	2	0.25	74.21
BERT$_{\text{base}}$	Translated ReLU	2.5	0.25	74.28
BERT$_{\text{base}}$	Smooth K2 Loss	3	0.25	75.75
BERT$_{\text{base}}$	Smooth K2 Loss	2.5	0.25	75.89
BERT$_{\text{base}}$	Smooth K2 Loss	2	0.25	76.03
RoBERTa$_{\text{base}}$	Translated ReLU	2	0.25	74.00
RoBERTa$_{\text{base}}$	Translated ReLU	1.5	0.25	74.11
RoBERTa$_{\text{base}}$	Translated ReLU	1	0.25	74.28
RoBERTa$_{\text{base}}$	Smooth K2 Loss	2.5	0.25	75.89
RoBERTa$_{\text{base}}$	Smooth K2 Loss	3	0.2	75.90
RoBERTa$_{\text{base}}$	Smooth K2 Loss	3	0.25	76.04
Implementation on Contrastive Learning PLMs				
Nomic v1	Smooth K2 Loss	3.5	0.2	82.76
Nomic v1	Smooth K2 Loss	2.5	0.2	82.79
Nomic v1	Smooth K2 Loss	2	0.2	82.82
Nomic v1	Smooth K2 Loss	3	0.2	82.83
Jina v2	Smooth K2 Loss	3	0.15	83.51
Jina v2	Smooth K2 Loss	3	0.2	83.54
Jina v2	Smooth K2 Loss	3.5	0.2	83.55
Jina v2	Smooth K2 Loss	4	0.2	83.55

4.5 Ablation Studies

In section **??**, we initially demonstrated the effectiveness of our regression framework by comparing the performance of models utilizing both classification-based and regression-based strategies for multi-category STS tasks. Then, we elucidated the significance of zero-gradient buffer zones by evaluating the performance of models when selecting either Translated ReLU or L1 Loss, and Smooth K2 Loss or MSE Loss as the loss function. These comparisons directly align with the three core innovations of this paper and fulfill the role of ablation experiments.

Here, we extend our ablation study by evaluating our network architecture, as depicted in Figure **??**. Specifically, we seek to determine the necessity of concatenating $\mathbf{u}, \mathbf{v}$, and their element-wise difference $|\mathbf{u} - \mathbf{v}|$ in the final linear layer of the model. To this end, we employ both BERT and RoBERTa under the same experimental conditions outlined in section **??**, with the results presented in Table **??**.

The findings indicate that the concatenation method $(\mathbf{u}, \mathbf{v}, |\mathbf{u} - \mathbf{v}|)$ is the most effective for both PLMs, thus further validating the rationale behind our proposed scheme.

Table 6: Average Spearman's correlation scores obtained by models on seven STS tasks with different concatenation methods in the final linear layer of our Siamese neural network architecture.

PLM	Concatenation	Spearman
BERT$_{\text{base}}$	$(\mathbf{u}, \mathbf{v})$	53.30
BERT$_{\text{base}}$	$(\lvert\mathbf{u} - \mathbf{v}\rvert)$	54.84
BERT$_{\text{base}}$	$(\mathbf{u}, \mathbf{v}, \lvert\mathbf{u} - \mathbf{v}\rvert)$	76.03
RoBERTa$_{\text{base}}$	$(\mathbf{u}, \mathbf{v})$	60.99
RoBERTa$_{\text{base}}$	$(\lvert\mathbf{u} - \mathbf{v}\rvert)$	59.10
RoBERTa$_{\text{base}}$	$(\mathbf{u}, \mathbf{v}, \lvert\mathbf{u} - \mathbf{v}\rvert)$	76.04

5 Conclusion

In this paper, we propose an innovative regression framework and develop two simple yet efficacious loss functions: Translated ReLU and Smooth K2 Loss, to address multi-class STS tasks. Compared to traditional classification approaches, our regression modeling strategy effectively captures the progressive relationships between categories, thereby achieving superior performance while reducing the parameter count in the model's output layer.

Further empirical evidence demonstrates that our method can also be combined with leading contrastive learning models, leveraging fine-grained annotated data to further enhance their performance. Moreover, this approach proves to be more advantageous than continued fine-tuning through contrastive learning, both in terms of performance gains and computational efficiency.

To support further research, we have made our code and model checkpoints publicly available.

6 Limitations

Due to the lack of baselines and computational resource constraints, the experiments in this paper primarily focus on encoder-only discriminative models, rather than recently advanced generative pre-trained models (e.g., LLaMA [**?**], Mistral [**?**]). However, it is important to emphasize that, compared to mainstream generative PLMs, the models we selected—BERT, RoBERTa, Jina Embeddings v2, and Nomic Embed v1—have significantly fewer parameters. This results in higher inference efficiency, which is particularly advantageous in large-scale information retrieval and text clustering scenarios.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Key R&D Program of China (No. 2022YFB3103100).

A Data Filtering Method

As mentioned in section **??**, before applying the STS-B and SICK-R training sets for model updates, we implemented strict data filtering to ensure that no sentence pairs present in the test sets would appear in the fine-tuning corpus.

To elaborate on this process, we first take the SICK-R dataset as an example to illustrate the standard format of STS datasets. As shown in Table **??**, each sample consists of two text strings, "sentence 1" and "sentence 2," along with a floating-point number "score" that indicates the semantic similarity between them. We denote these as "s1," "s2," and "r," respectively.

Then, for any sentence pair $(s1_i, s2_i, r_i)$ within the STS-B or SICK-R training set, if a sample $(s1_j, s2_j, r_j)$ exists in the test sets of STS12-16, STS-B, or SICK-R such that $s1_i = s1_j$ and $s2_i = s2_j$, or $s1_i = s2_j$ and $s2_i = s1_j$, we treat them as duplicates and remove the corresponding sentence pair from the training data. It should be noted that the entire process is conducted without any modifications to the test sets.

This filtering mechanism is stringent, as we do not take into account whether r_i and r_j are equal. In other words, as long as a sentence pair appears in both the training and test sets, it is removed from the training corpus, regardless of whether the similarity scores are identical. Under this protocol, even within the SICK-R dataset itself, there are instances where samples from the training and test sets overlap. Examples in Table **??** illustrate such cases. The goal of this approach is to maximize the model's generalization ability.

Table 7: Samples from the SICK-R training and test sets. In these samples, text pair duplication occurs. Thus, the corresponding training samples are removed from the fine-tuning corpus used in section **??**.

sentence 1	sentence 2	score
	Sentence pairs in the SICK-R training set	
A man in a blue jumpsuit is courageously performing a wheelie on a motorcycle	The man is doing a wheelie with a motorcycle on ground which is mostly barren	4.1
The tan dog is watching a brown dog that is swimming in a pond	A pet dog is standing on the bank and is looking at another brown dog in the pond	4.3
	Sentence pairs in the SICK-R test set	
The man is doing a wheelie with a motorcycle on ground which is mostly barren	A man in a blue jumpsuit is courageously performing a wheelie on a motorcycle	3.7
A pet dog is standing on the bank and is looking at another brown dog in the pond	The tan dog is watching a brown dog that is swimming in a pond	3.6