SEAVER: Attention Reallocation for Mitigating Distractions in Language Models for Conditional Semantic Textual Similarity Measurement

Anonymous ACL submission

Abstract

Conditional Semantic Textual Similarity (C-STS) introduces specific limiting conditions to the traditional Semantic Textual Similarity (STS) task, posing challenges for STS models. Language models employing cross-encoding demonstrate satisfactory performance in STS, yet their effectiveness significantly diminishes in C-STS. In this work, we argue that the failure is due to the fact that the redundant information in the text distracts language models from the required condition-relevant information. To alleviate this, we propose *SElf-Augmentation Via SElf-Reweighting* (SEAVER), which, based solely on models' internal attention and without the need for external auxiliary information, adaptively reallocates the model's attention weights by emphasizing the importance of condition-relevant tokens. On the C-STS-2023 test set, SEAVER consistently improves performance of all million-scale fine-tuning baseline models (up to around 3 points), and even surpasses performance of billion-scale few-shot prompted large language models (such as GPT-4). Our code is available at https://github.com/NLP-LEE/SEAVER.

1 Introduction

Semantic Textual Similarity (STS) has been a cornerstone task in natural language processing fields for years (Agirre et al., 2014, 2015, 2016; Cer et al., 2017; Abdalla et al., 2021), which aims to measure the semantic similarity between two sentences. With the emergence of pre-trained language models (Devlin et al., 2018; Liu et al., 2019; Brown et al., 2020; Raffel et al., 2020), the STS task seems to have been almost solved. However, STS is an inherently ambiguous task (Wang et al., 2023b), for the varying aspects that can influence sentence similarity, unconditionally measuring this similarity is irrational and unexplainable. To solve the ambiguity of STS task itself, Deshpande et al. (2023) proposed a novel task called Conditional Semantic Textual Similarity (C-STS), which incorporates specific conditions to highlight fine-grained aspects of interest in sentence pair similarity assessment (as shown in Figure 1), enables a more grounded, precise and multi-faceted evaluation.

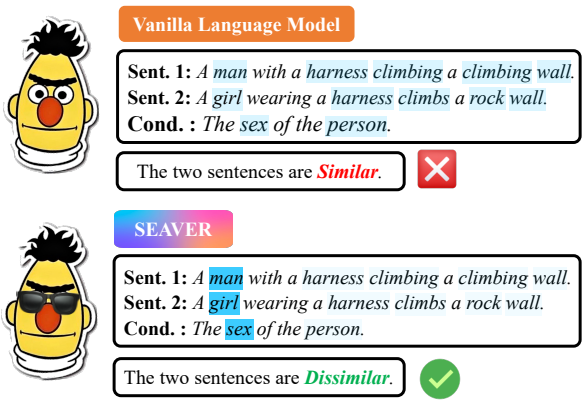

Figure 1: A straightforward example illustrating the distraction in language models, SEAVER is able to softly filter out irrelevant information, thereby focusing the model's attention on condition-relevant tokens.

Given that C-STS introduces additional complexity into STS, researchers have explored various mainstream models, attempting to transfer them from STS to C-STS (Liu et al., 2019; Reimers and Gurevych, 2019; Deshpande et al., 2023). However, the results obtained have been less than satisfactory. State-of-the-art STS language models (hereafter referred to as STS models), such as SimCSE (Gao et al., 2021), achieve only relatively low performance in C-STS even after fine-tuning on the C-STS dataset. More notably, even few-shot prompted large language models perform poorly in C-STS. This prompts us to ask: *What causes the state-of-the-art models in STS to fail in C-STS?*

Previous work confirms that redundant objects in data can distract models, leading to suboptimal performance, a phenomenon widely discussed in the visual domain (Wang et al., 2023a; You et al., 2023). However, this issue also exists in the text domain, where pre-trained language models often

Method	Encoder Type	Additional Part	#CM	#FF	Reweight	Application Field
Vanilla LMs (Gao et al., 2021)	cross-encoder	none	1	1	✗	text-only
PerceiverIO (Jaegle et al., 2021)	cross-encoder	cross-attn module	3	1	✓	multimodal
AbSViT (Shi et al., 2023)	bi-encoder	feedback network	2	2	✓	visual & multimodal
SEAVER (Ours)	**cross-encoder**	**none**	**1**	**1**	✓	**text-only**

Table 1: Comparison of related work. "#CM" and "#FF" represent the number of computational module types required for a single feedforward pass and the number of feedforward passes needed for one prediction, respectively.

extract excessive potential semantic information (Hewitt and Manning, 2019), most of which is irrelevant to the task. The design of STS inherently overlooks this issue, but C-STS has prompted a rethinking of redundancy in the text domain.

As shown in Figure 1 (top), the two sentences displayed differ only in the gender-specific aspect (condition-relevant), while all other aspects (condition-irrelevant) are semantically identical. However, since the dissimilar but condition-relevant aspect occupies a relatively small proportion within the sentences, the abundance of similar but condition-irrelevant aspects vastly exceeds the required judgment area restricted by the condition in the sentences. Due to the unconditional design of STS, the STS models fine-tuned on C-STS still tend to largely rely on the excessive similar but condition-irrelevant semantic features, ignoring the dissimilar but condition-relevant aspects that truly require the model's focus. This leads to their attention being largely distracted. As a result, the models tend to mistakenly perceive the sentences as highly similar, and this inclination is difficult to eliminate through simple fine-tuning.

Given the aforementioned observations, we argue that the excessive semantic features extracted by language models, which, in turn, distracts their attention, is the key reason for the failure of STS models in C-STS. As similar phenomena have been observed in the fields of visual and multimodal, researchers in these fields attempt to mitigate such distractions using *reweighting* strategies (Jaegle et al., 2021; Shi et al., 2023).

Inspired by the *reweighting* strategy, we propose a novel method that directly extracts the internal condition-sentence cross-attention submatrices, which contain condition-sentence correlations, from the STS model. Utilizing these submatrices, we construct reweighting matrices to emphasize the importance of condition-sentence correlations in attention allocation. Considering the preservation of the overall semantic integrity, the *reweighting* results serve as an *augmentation* signal to enhance the original output hidden states, explicitly directing the model to focus more on condition-relevant tokens (as shown in Figure 1). Since our proposed method solely utilizes internal attention information, we have named it *SElf-Augmentation Via SElf-Reweighting* (SEAVER).

Retaining an architecture that is relatively consistent with that of the pre-trained language model, SEAVER exhibits the capability to outperform all fine-tuning baselines on the C-STS-2023 test set (Deshpande et al., 2023). Remarkably, with a significantly smaller parameter scale, it also surpasses the performance of most few-shot prompted large language models, highlighting its significant potential in advancing C-STS measurement.

2 Related Work

Excessive features extracted by Language Models. There is substantial evidence indicating that throughout the pre-training, language models learn not only contextualized text representations, but also a grasp of grammar (Vig, 2019), syntax (Hewitt and Manning, 2019), even commonsense (Davison et al., 2019) and world knowledge (Petroni et al., 2019; Wang et al., 2020).

However, the semantic information mentioned above is general-purpose and unconditional. Thus, for C-STS, which emphasize the conditional constraints on sentences and focus on more fine-grained aspects, the excessive information can, in turn, distract the language model's attention.

Conditional Reweighted Feedforward. Tasks similar to C-STS (Deshpande et al., 2023) find more discussions in vision and multimodal fields (Deng et al., 2009; Carrasco, 2011; Li, 2014; Antol et al., 2015). In these contexts, a specific condition is essential for directing the model's focus towards objects that are relevant to the given condition.

Previous work employing such methods has yielded effective results. PerceiverIO (Jaegle et al., 2021) introduced multiple cross-attention modules to compute the relevance to reweight the output

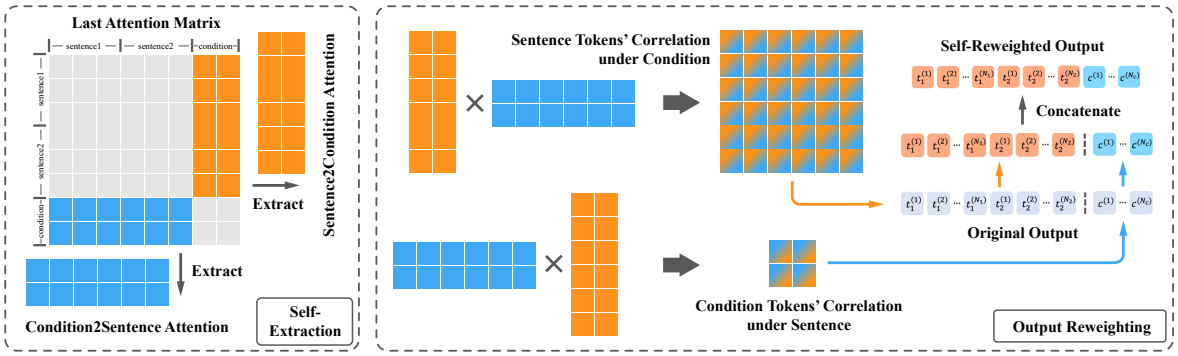

Figure 2: Self-Reweighting flow (from left to right). (i) Self-Extraction: extract attention submatrix, which represents the interaction between the sentence and the condition. (ii) Output Reweighting: compute attention reallocation matrices, serving to reweight the original output hidden states of the sentence and the condition, respectively, then concatenate them, culminating in the acquisition of a self-reweighted output hidden state.

tokens, which were directly used for prediction. Conversely, AbSViT (Shi et al., 2023) proposed a feedback mechanism to feed the relevance computed during the first feedforward phase back to the preceding modules, then the second feedforward were conducted for prediction.

Moreover, these methods only apply *reweighting* to visual features, and the textual component (if present in the task) is often represented only in a short-form indicative manner and does not participate in reweighting. Due to the inherent differences in information density between textual and visual data (He et al., 2022), such *reweighting* strategies for visual features do not meet the requirements of C-STS. As shown in Table 1, inspired by previous work, we design a *reweighting* strategy better suited for C-STS, enabling a more efficient computing flow and a more integrated computing structure.

3 Method

This section starts with *Self-Reweighting*, which directly extracts condition-sentence cross-attention submatrices to reweight the outputs (Section 3.1), then we use the reweighted outputs to enhance the original outputs in a specific proportion (Section 3.2), namely *Self-Augmentation*.

3.1 Self-Reweighting

As is well known, when utilizing cross-encoding, we compute the attention of the concatenated sentence pair and the condition, which actually encapsulates multi-faceted information, encompassing both the *self-attention* of each input item and the *cross-attention* among input items.

Based on such observations, unlike previous attempts to introduce external auxiliary information or computational modules (Jaegle et al., 2021; Shi et al., 2023), we designed a novel method to *construct the reweighting matrix directly using the internal attention in the model*. As shown in Figure 2, to emphasize the condition-relevant information, we specifically extract the cross-attention between the sentences and the conditions from the whole attention matrix. Then we divide them into two distinct aspects of attention, namely *Sentence2Condition Attention* (**SCAttn**) and *Condition2Sentence Attention* (**CSAttn**), respectively. Here, $\mathbf{SCAttn} \in \mathbb{R}^{l_s \times l_c}$ and $\mathbf{CSAttn} \in \mathbb{R}^{l_c \times l_s}$, where l_s indicates the length of the concatenated sentence pair, and l_c indicates the condition length.

We use the extracted **SCAttn** as the condition-guided signal for sentences and **CSAttn** as the sentence-guided signal for conditions. Utilizing these, we calculate their similarities to construct the reweighting matrices for sentences and conditions, respectively. This reallocates attention by integrating sentence and condition information with each other, which are computed as

$$\mathbf{W}_S = \mathrm{softmax}(\mathbf{SCAttn} \cdot \mathbf{CSAttn}) \quad (1)$$

$$\mathbf{W}_C = \mathrm{softmax}(\mathbf{CSAttn} \cdot \mathbf{SCAttn}), \quad (2)$$

where $\mathbf{W}_S \in \mathbb{R}^{l_s \times l_s}$ indicates the reweighting matrix for sentences and $\mathbf{W}_C \in \mathbb{R}^{l_c \times l_c}$ indicates the reweighting matrix for conditions.

Applying the obtained reweighting matrices $\mathbf{W}_S$ and $\mathbf{W}_C$, we perform Self-Reweighting on the truncated model outputs, which can be computed as

$$\mathbf{RO}_S = \mathbf{W}_S \cdot \mathbf{O}[0 : (l_s - 1)] \quad (3)$$

$$\mathbf{RO}_C = \mathbf{W}_C \cdot \mathbf{O}[l_s : (l_s + l_c)], \quad (4)$$

where $\mathbf{O} \in \mathbb{R}^{l \times d}$ indicates the last hidden state of the language model, which we subsequently refer

to as the original output in the following text. l and d represent the length of the concatenated input (comprising the sentence pair and the condition) and the dimension of the language model's hidden state, respectively. Here we represent the i-th token of sentence k ($k \in \{1, 2\}$) as $t_k^{(i)}$. $\mathbf{RO}_S \in \mathbb{R}^{l_s \times d}$ and $\mathbf{RO}_C \in \mathbb{R}^{l_c \times d}$ represent the reweighted output of the sentence pair and the condition, respectively.

After acquiring the reweighted outputs for both sentences and conditions, we then concatenate them to form the concatenated reweighted outputs $\mathbf{RO} \in \mathbb{R}^{l \times d}$, where $\mathbf{RO}$ indicates the concatenated reweighted output, which is of the same size with the original output $\mathbf{O}$. Then, we utilize the reweighted (attention reallocated) output $\mathbf{RO}$ as an augmentation signal to perform the Self-Augmentation as described in Section 3.2.

Furthermore, it is important to note that the reweighting matrices for attention reallocation are derived directly from the attention matrices returned by the last layer of the language model. Since this does not introduce an external information, we refer to this process as *Self-Reweighting*.

3.2 Self-Augmentation

We consider the multi-head self-attention mechanism of the language model, which ultimately yields H attention matrices, where H is the number of attention heads. Here, we refer to the reweighted output obtained after applying the reweighting matrices constructed from the attention matrix returned by the i-th attention head as $\mathbf{RO}_i$. Following a method similar to that used in Transformers for processing outputs from multiple attention heads (Vaswani et al., 2017), we concatenate these H reweighted outputs. Subsequently, they are projected through a projection matrix to match the dimension of a single reweighted output, which can be computed as

$$\mathbf{RO} = [\mathbf{RO}_1; \mathbf{RO}_2; ...; \mathbf{RO}_H] \cdot \mathbf{W}_o, \quad (5)$$

where $\mathbf{W}_o \in \mathbb{R}^{Hd \times d}$ indicates the projection matrix. To be more specific, the $\mathbf{RO}$ here indicates the projected reweighted output. Each $\mathbf{RO}_i$ is computed through Section 3.1, where it should be noted that the $\mathbf{RO}$ in Section 3.1 denotes the case for a single attention head.

We utilize the final reweighted output $\mathbf{RO}$ as an augmentation signal, aimed at enhancing parts of the original output $\mathbf{O}$ where there is a significant semantic association between the sentence pair and the condition. To achieve this, we perform a weighted addition of the augmentation signal $\mathbf{RO}$ with the original output $\mathbf{O}$. This results in the self-augmented output, which is then utilized for predicting similarity, which can be computed as

$$\mathbf{AO} = \mathbf{RO} + \alpha\mathbf{O}, \quad (6)$$

where $\mathbf{AO} \in \mathbb{R}^{l \times d}$ indicates the self-augmented output and $\alpha \geq 0$ denotes the hyperparameter that controls the ratio between the weight of reweighted output $\mathbf{RO}$ and the original output $\mathbf{O}$, which is discussed in detail in Section 4.2.

The overall architecture of the model is as depicted in Figure 3, where the final regressor is a single-hidden-layer MLP structure for scoring.

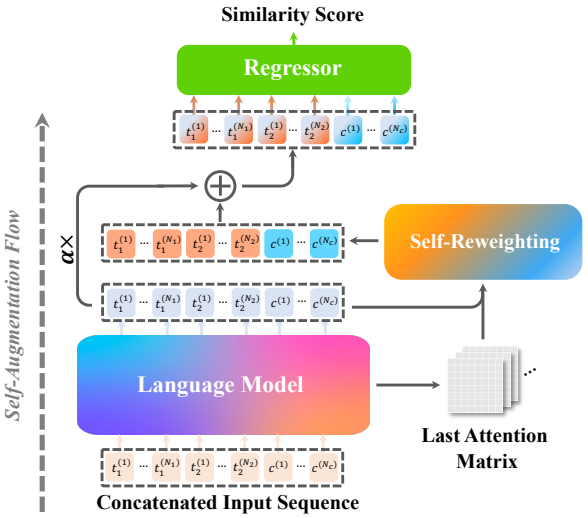

Figure 3: Overall architecture of our proposed SEAVER. A self-augmented output is derived through the addition of the self-reweighted output to the original output (scaled by a factor of α). This self-augmented output is subsequently fed into a simple regressor (a single-hidden-layer MLP), predicting the semantic similarity.

4 Experiments

In this section, we first demonstrate the attention reallocation effect of SEAVER (Section 4.1). Subsequently, we provide a detailed quantitative analysis to discuss the improvements provided by SEAVER (Section 4.2). Finally, we present separate ablation studies for the Self-Reweighting (Section 4.3) and Self-Augmentation (Section 4.4) in SEAVER.

Dataset. In this study, we employ C-STS-2023 dataset collected by Deshpande et al. (2023) for training and testing, which consists of quadruples, formatted as (`sentence1`, `sentence2`,

Sentence 1	Sentence 2	Condition	Output
A boy is in midair doing a skateboard trick at a skate park while two women and a toddler walk behind him.	A boy in yellow pants and a blue shirt is rollerblading on the side of his black skates.	The type of skating.	w/o: 4.00 w/ : 1.46 Label: 1.00
Two people are near a wooden building wearing backpacks.	A couple of people working around a pile of rocks.	The number of people.	w/o: 2.60 w/ : 4.62 Label: 5.00

Table 2: Two cases from the C-STS-2023 validation set. "Output" refers to the predicted and the ground-truth similarity, where the notation "w/o" represents the prediction from the baseline model, and "w/" denotes the prediction from our proposed SEAVER. More cases are available in Appendix A.1.

`condition`, `label`). In which `label` represents the level of similarity between `sentence1` and `sentence2` under `condition`, converted into a Likert scale (Likert, 1932) with values ranging from 1 to 5, which is common with semantic textual similarity tasks (Agirre et al., 2013).

Experimental Setup. We conduct a comparative analysis between various baselines and our proposed SEAVER, which can be categorized into:

(i) **Fine-tuning** baselines, which are fine-tuned on the entire training partition. We select RoBERTa (Liu et al., 2019) and SimCSE (Gao et al., 2021) as our language model baselines, encompassing both the `base` and `large` scales. Additionally, we have considered top-notch works that possess design principles analogous to SEAVER, as detailed in Section 2, as baseline models. These include AbS-LM and PerceiverIO (Jaegle et al., 2021), where AbS-LM represents a modified AbSViT (Shi et al., 2023) for C-STS, with its ViT backbone replaced by RoBERTa and SimCSE (denoted as AbS-RoBERTa and AbS-SimCSE, respectively). For PerceiverIO, we selected the version of the model pre-trained exclusively for text tasks.

(ii) **Prompting** baselines, which refer to general-purpose large language models, are recognized for their few-shot learning capabilities. We select Flan-T5 (Wei et al., 2021), GPT-J (Wang and Komatsuzaki, 2021), GPT-3.5 (Brown et al., 2020), and GPT-4 (Achiam et al., 2023) as our baselines.

It is important to note that due to observed variances in experimental results across different models of GPUs, to ensure reproducibility, all experiments were conducted on a single RTX A5000. More details are available in Appendix A.2.

4.1 Dilution Effect and SEAVER Mitigation

In Table 2, the predictions from the baseline model are higher and lower in comparison to the ground-truth, respectively, while those from SEAVER align more closely with the ground-truth.

To elucidate the attention allocation mechanism of the baseline model in C-STS, and to understand the reasons behind the baseline model's prediction failures as well as the success of SEAVER. As illustrated in Figure 4, we extracted and averaged the attention matrices from the last layer of the baseline model and the Self-Reweighting weights for the sentence part in SEAVER.

Since the input sequence consists of concatenated sentence and condition, SEAVER includes separate reweighting matrices that affect both the sentence and the condition respectively. However, considering that the condition itself serves to impose constraints on the sentence. In this section, to more intuitively understand how SEAVER reweights the sentence based on the condition, we only display the reweighting matrix that acts on the sentence part (Figure 4 (right)).

In Figure 4 (left), it is observable that in the baseline model, the required Region of Interest (RoI) does not receive additional attention. We also observed that the required RoI occupies only a small proportion within the sentence, with the remaining parts involved in attention computation predominantly consisting of numerous condition-irrelevant tokens, which, after being normalized by the softmax function, dilute the impact of condition-relevant features on the final prediction. We have named this the *Dilution Effect*.

After applying our proposed SEAVER method, we observe from Figure 4 (right) that the reweighting matrix exhibits distinct emphasized regions (darker in color) and suppressed areas (lighter in color). This refocuses attention on the condition-

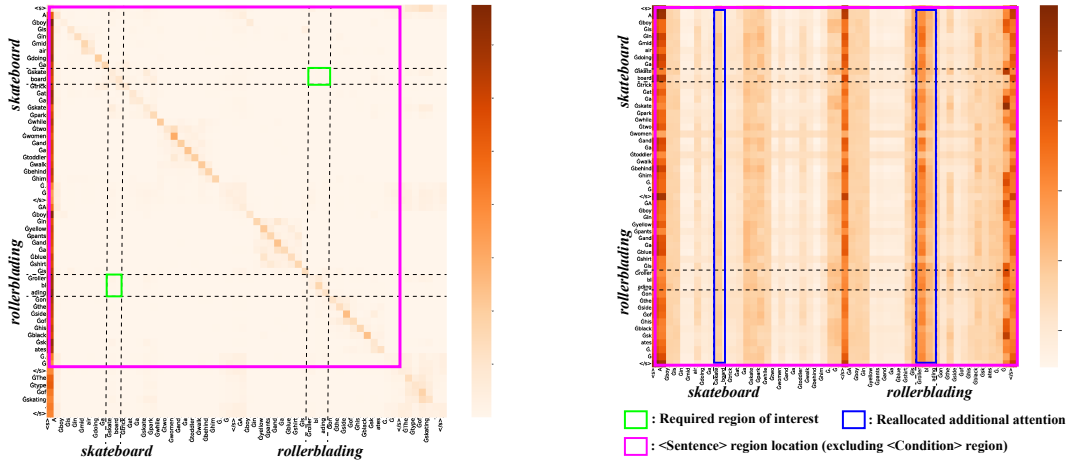

Figure 4: Average attention matrix (left: obtained from the baseline model RoBERTa-base) and Reweighting matrix specifically for **sentence parts'** attention reallocation (right: obtained from SEAVER) of **the first-row case** presented in Table 2. The darker the color, the larger the score. The words on the horizontal and vertical axes are complete words formed by concatenating the tokens at corresponding positions. We have outlined the attention regions involved. An enlarged version can be find in Appendix A.1 for a clearer display.

relevant tokens. For instance, for the first case in Table 2, the emphasized regions of the reweighting matrix make the model concentrate more on tokens related to `the type of skating`, such as `skateboard` and `rollerblading`. Consequently, compared to the baseline model, applying SAVER successfully reallocates more attention to the condition-relevant aspects, mitigating distractions within the model.

4.2 Quantitative Results and Analysis

We initially conduct fine-tuning experiments using the entire training set of the C-STS-2023 dataset. The quantitative results are shown in Table 3. More details are available in Appendix A.3.

In Table 3, RoBERTa has been fine-tuned directly on the C-STS-2023 dataset following pre-training. In contrast, before being fine-tuned on the C-STS-2023 dataset, SimCSE has already been fine-tuned on unconditional STS datasets. It's observable that our proposed SEAVER can bring stable performance improvements to these two baseline language models of different scales.

Furthermore, we also compared the performance of SEAVER with that of novel related works possessing analogous design principles on the C-STS task (AbS-LM and PerceiverIO). The results indicate that the two approaches, analogous in design to ours, performed poorly on the C-STS task, even falling significantly short of the performance of vanilla language models. The reasons for this underperformance are as follows:

Model	#Param.	Spear. ↑	Pears. ↑
PerceiverIO	203M	1.26	1.32
RoBERTa	125M	39.07	39.05
AbS-RoBERTa	139M	8.58	8.04
SEAVER RoBERTa	**132M**	**41.36**	**41.05**
RoBERTa	355M	40.40	40.78
AbS-RoBERTa	406M	-3.48	-1.84
SEAVER RoBERTa	**372M**	**43.45**	**43.60**
SimCSE	125M	38.56	39.00
AbS-SimCSE	139M	6.47	6.28
SEAVER SimCSE	**132M**	**39.59**	**39.30**
SimCSE	355M	42.28	42.40
AbS-SimCSE	406M	9.55	9.20
SEAVER SimCSE	**372M**	**43.83**	**43.81**

Table 3: Fine-tuning results in Spearman and Pearson correlation (scaled by 100) on the C-STS-2023 test set. Highlighted rows indicate optimal performance with the best-configured α within a series.

Intrusive reweighting strategy disrupts the capability for attention allocation. AbS-LM retains parts of the original Language Model (LM) and introduces feedback information in an intrusive manner (i.e., directly reweighting the value part of attention in LMs based on the similarity between condition embeddings and extracted features). However, this intrusive feedback method not only introduces a significant number of additional parameters, leading to training instability, but also disrupts the internal information of pre-trained LMs, resulting in failure on the C-STS task.

Simple cross-attention modules struggle to meet the demands of C-STS. Although PerceiverIO introduces cross-attention modules more in line with the C-STS task setting compared to Vanilla LMs, it lacks the powerful semantic understanding inherent to pre-trained language models, thereby only performing superficial similarity measurements on texts without capturing deeper semantic information, which is crucial for C-STS.

In contrast to these methods, SEAVER utilizes a residual connection-style non-intrusive approach to reallocate attention by emphasizing the internal condition-relevant information within its attention matrices, thereby focusing more on condition-relevant aspects. This results in a minimal increase in parameters without introducing any additional cross-attention modules, further validating the effectiveness and efficiency of SEAVER.

Model	0-shot ↑	2-shot ↑	4-shot ↑
Flan-T5-base	11.3	9.1	10.7
Flan-T5-large	11.1	12.3	12.8
GPT-J	7.4	1.1	2.0
GPT-3.5	15.0	16.6	15.5
GPT-4	39.3	42.6	43.6
Our fine-tuned model (w/ the best performance)			
†**SEAVER SimCSE (372M)**			**43.8**

Table 4: Zero-shot and few-shot prompted results on the C-STS-2023 test set using Spearman's correlation. † indicates fine-tuning on the entire training set.

Additionally, we compared the performance of SEAVER with that of zero-shot and few-shot prompted large language models on the C-STS-2023 test set. The performance of the zero-shot and few-shot prompted large language models, as presented in Table 4, represent the best results obtained after prompting using various prompts as applied by Deshpande et al. (2023).

As shown in Table 4, it is evident that despite a substantial difference in the number of parameters between our selected model (372M) and large language models such as GPT-J (6B), GPT-3.5 (175B), and GPT-4 (even larger than GPT-3.5), the best performance of SEAVER, still surpasses the optimal performance achieved by large language models. Furthermore, as the process of zero-shot and few-shot prompting in large language models also constitutes cross-encoding, this further confirms the superiority of SEAVER in cross-encoding models.

4.3 Self-Reweighting Impact Analysis

Given that Self-Reweighting extracts the condition-sentence cross-attention submatrices, we now commence with the random selection of two non-overlapping submatrices from the attention matrix to further confirm the effectiveness of Self-Reweighting. These submatrices, similar to those in the Self-Reweighting configuration, are symmetrically positioned relative to the main diagonal of the attention matrix, and are utilized as the weights for reweighting, a process we have termed *Random-Augmentation*, which yielded the following results:

Model	#Param.	Spear. ↑	Pears. ↑
RoBERTa	125M	39.07	39.05
+Rand-Aug w/o orig.	132M	38.00	37.57
+Rand-Aug w/ 1*orig.	132M	37.78	37.56
+Rand-Aug w/ 2*orig.	132M	37.48	37.26
+Rand-Aug w/ 3*orig.	132M	35.00	35.48
RoBERTa	355M	40.40	40.78
+Rand-Aug w/o orig.	372M	40.93	40.83
+Rand-Aug w/ 1*orig.	372M	38.86	38.91
+Rand-Aug w/ 2*orig.	372M	40.83	40.95
+Rand-Aug w/ 3*orig.	372M	40.41	40.26
SimCSE	125M	38.56	39.00
+Rand-Aug w/o orig.	132M	37.37	37.11
+Rand-Aug w/ 1*orig.	132M	37.52	37.08
+Rand-Aug w/ 2*orig.	132M	37.39	37.43
+Rand-Aug w/ 3*orig.	132M	37.86	37.96
SimCSE	355M	42.28	42.40
+Rand-Aug w/o orig.	372M	41.16	41.01
+Rand-Aug w/ 1*orig.	372M	40.08	39.79
+Rand-Aug w/ 2*orig.	372M	43.07	43.12
+Rand-Aug w/ 3*orig.	372M	42.60	42.75

Table 5: Fine-tuning results of Random-Augmentation on the C-STS-2023 test set. Highlighted rows indicate declined performance within a series. "+Rand-Aug w/ α*orig." denotes the addition of the Random-Reweighting signal to the original output (scaled by a factor of α), and "w/o" is equivalent to $\alpha = 0$.

From Table 5, it can be observed that Random-Augmentation does not enhance the performance of the language model on the C-STS task in the majority of cases. However, in some instances, slight improvements over the baseline were observed, attributable to four primary reasons:

(i) The introduction of additional parameters (albeit minimal) allowed for minor gains. The inclusion of new parameters in the model can subtly enhance its performance by providing more flexibility in adapting to the data.

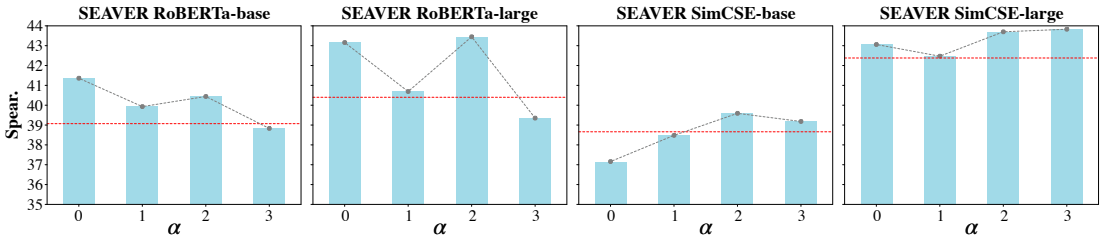

Figure 5: Spearman's correlation of SEAVER under different settings of α. The red dashed line represents the performance of the corresponding fine-tuning baseline language model. Detailed values can be found in Table 7.

(ii) The randomly sampled submatrices inevitably encompass parts of the condition-sentence cross-attention submatrices from the attention matrix. Therefore, compared to the unenhanced baseline model, this inclusion also contributes to a partial gain.

(iii) As α increases, the proportion of the original signals extracted by the language model is amplified, thereby diminishing the impact of the Random-Augmentation signal. A detailed discussion regarding the impact of α is provided in Section 4.4.

(iv) Random-Augmentation introduces a certain amount of noise into the fine-tuning process. Several studies (Zhang et al., 2020; Wu et al., 2022) have indicated that the introduction of such noise can reduce the gap between pre-training and fine-tuning tasks, thereby having a positive impact on fine-tuning.

Nevertheless, it is evident that these gains do not match the improvements afforded by SEAVER of extracting specific cross-attention submatrices through Self-Reweighting. This further corroborates the effectiveness of the Self-Reweighting strategy's intuitively designed rationale and also demonstrates that the improvements introduced by SEAVER are not merely the result of increased parameters and training perturbations.

4.4 Self-Augmentation Ratio Analysis

To explore optimal performance of SEAVER, we configured 4 different Self-Augmentation Ratios α on various versions of SEAVER as shown in Figure 5. It is clear that there exists an easily identifiable, optimal configuration of α that enables the best synergy between the model's original output and the augmentation signal, ensuring that SEAVER consistently outperforms the baseline model.

Additionally, to analyse the impact of α. As specified in Equation 6, a larger α increases the proportion of the original output's influence on the final prediction. When $\alpha = 0$, the final prediction relies solely on the augmentation signal. As $\alpha \to +\infty$, it depends exclusively on the original output (degenerates to the baseline model).

It can be observed that the optimal configuration of α is not zero in most cases, confirming that, in addition to directly condition-relevant features, the preservation of the overall semantics, which is largely provided by the original output, also plays a crucial role. Therefore, this is the rationale for using the Self-Reweighting output as an augmentation signal to the original output, rather than as the sole component utilized for prediction.

Meanwhile, the optimal configuration of α varies across models of different scales and training methods. We note that α represents a form of trade-off between the model's intrinsic sentence understanding ability and the degree of need for attention reallocation. Models with stronger sentence understanding, such as RoBERTa-large, typically require a larger α value compared to RoBERTa-base, i.e. models with higher intrinsic sentence understanding have less need for attention reallocation through the Self-Reweighting output to mitigate distraction. More details are available in Appendix A.4.

5 Conclusion

In this work, we argue that the reason for the subpar performance of language models in C-STS is attributed to the dilution effect: The excessive general-purpose but condition-irrelevant features distract language models' attention from the specific, condition-relevant features that occupy a relatively small proportion in the sentence. However, mitigating this distraction through mere fine-tuning is challenging. To address this, we propose SEAVER, which reallocates the model's attention weights based on specific conditions using its internal information. On the C-STS-2023 test set, our method outperforms all types of baseline models.

Limitations

Although the application of SEAVER can bring stable performance improvements to models using cross-encoding, proving its feasibility, due to concerns about the method's complexity, SEAVER only involves extracting relevant attention scores from the last layer of the language model and calculating the semantic correlation between sentences and conditions. This results in the extracted relevance reflecting more on the independent semantic features of the last layer, which does not significantly enhance performance.

In this study, experiments have demonstrated that small models applying our proposed method can achieve performance surpassing that of few-shot prompted large language models. However, due to limitations in computational resources, we did not apply our method to larger scale models.

Future work can focus on the comprehensive utilization of semantic relevance captured in other layers of the model, as well as that of the last layer and other layers. Furthermore, the adoption of a learned adaptive approach to make models focus more on condition-relevant semantic features of each layer can be considered. This would enable adaptive amplification of a certain number of semantic features according to the complexity of different sentences, thereby achieving more efficiency and satisfactory performance improvements. Additionally, future work should consider extending this method to larger scale models to explore more of the method's potential.

Ethical Considerations

It is widely acknowledged that language models are capable of generating predictions that exhibit bias. This issue becomes especially pronounced when the input sentences possess sensitive characteristics. While strategies such as data cleaning can alleviate these problems, they do not offer a complete solution. In light of some potential issues, this study advocates for usage under research purposes. Appropriate care should thus be taken when applying such approaches for any non-research purpose (e.g. in user-oriented applications).

In this study, our use of existing artifacts is consistent with their intended purposes. All the datasets and models used in this work are publicly available. RoBERTa-* models have MIT license[1]. Flan-T5-* and PerceiverIO models have Apache-2.0 license[2]. The remaining open-source models and datasets used, due to the lack of explicit licensing declarations, have all been credited with their sources in Appendix A.2 in this paper.

[1] https://choosealicense.com/licenses/mit

[2] https://www.apache.org/licenses/LICENSE-2.0

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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 Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, 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, 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.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Marisa Carrasco. 2011. Visual attention: The past 25 years. *Vision research*, 51(13):1484–1525.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Ameet Deshpande, Carlos E Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. 2023. Csts: Conditional semantic textual similarity. *arXiv preprint arXiv:2305.15093*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. 2021. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*.

Zhaoping Li. 2014. *Understanding vision: theory, models, and data*. Oxford University Press (UK).

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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.

Baifeng Shi, Trevor Darrell, and Xin Wang. 2023. Top-down visual attention from analysis by synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2102–2112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Jing Wang, Peitong Li, Rongfeng Zhao, Ruyan Zhou, and Yanling Han. 2023a. Cnn attention enhanced vit network for occluded person re-identification. *Applied Sciences*, 13(6):3707.

Yuxia Wang, Shimin Tao, Ning Xie, Hao Yang, Timothy Baldwin, and Karin Verspoor. 2023b. Collective human opinions in semantic textual similarity. *Transactions of the Association for Computational Linguistics*, 11:997–1013.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2022. Noisytune: A little noise can help you finetune pretrained language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–685.

Haoran You, Yunyang Xiong, Xiaoliang Dai, Bichen Wu, Peizhao Zhang, Haoqi Fan, Peter Vajda, and Yingyan Celine Lin. 2023. Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14431–14442.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*.

A Appendix

A.1 Dilution Effect and SEAVER Mitigation

Additional cases, along with their corresponding attention matrices and Self-Reweighting weights, are provided in Table 8 and Figure 6, respectively. This enables a broader and deeper understanding of the dilution effect and SEAVER alleviation mentioned in Section 4.1. An enlarged version of Figure 4 can be find in Figure 7 and Figure 8.

It must be reiterated that the Self-Reweighting weights computed here reflect the reallocation of different features' intensities. That is, to enhance condition-relevant features and suppress condition-irrelevant features, it is necessary to adjust the intensity of the original features. Therefore, in the Self-Reweighting weights, there may be instances where the weights of features that are supposed to be enhanced are not as salient. This can occur not only due to the intrinsic learning quality of the model but also because the original intensity of certain features is already relatively strong, thus requiring less enhancement, and vice versa.

A.2 Implementation Details

The hyperparameter settings shown in Table 6 were determined to yield the best performance when evaluating our proposed SEAVER on the C-STS-2023 validation set. To maintain higher consistency with the baseline proposed by Deshpande et al. (2023), and to maximize the reproducibility of our experimental results, we set the torch seed to 42 in all our experiments.

As mentioned by Deshpande et al. (2023), the C-STS-2023 dataset used in this paper comprises a training set (11,342 examples), a validation set (2,834 examples), and a test set (4,732 examples), all consisting of English sentence examples.

All pre-trained parameters of the language models involved in the experiments are directly available on Hugging Face: RoBERTa-base[3], RoBERTa-large[4], SimCSE-base[5], SimCSE-large[6], and PerceiverIO[7]. In Table 3, we mention AbS-LM, which is a variant based on the AbSViT model that substitutes the ViT backbone with a language model. The original AbSViT model has also been made open source[8]. For GPT-3.5 and GPT-4, consistent with the experimental setup described by Deshpande et al. (2023), the related test results were obtained using the OpenAI API with the static model versions `gpt-3.5-turbo-0301` and `gpt-4-0314` during the experiments.

Configuration	Base	Large
Batch Size	64	64
Learning Rate	3e-5	1e-5
Weight Decay	0.1	0.1
Seed	42	42
Loss	MSE	MSE

Table 6: Hyperparameter sweep done for C-STS-2023 validation set for our proposed Self-Augmentation models. "Base" and "Large" represent the scale of our proposed Self-Augmentation models.

Model	#Param.	Spear. ↑	Pears. ↑
RoBERTa (Deshpande et al., 2023)	125M	39.07	39.05
SEAVER RoBERTa w/o orig.	**132M**	**41.36**	**41.05**
SEAVER RoBERTa w/ 1*orig.	132M	39.93	39.83
SEAVER RoBERTa w/ 2*orig.	132M	40.44	40.35
SEAVER RoBERTa w/ 3*orig.	132M	38.83	38.91
RoBERTa (Deshpande et al., 2023)	355M	40.40	40.78
SEAVER RoBERTa w/o orig.	372M	43.16	43.20
SEAVER RoBERTa w/ 1*orig.	372M	40.69	40.56
SEAVER RoBERTa w/ 2*orig.	**372M**	**43.45**	**43.60**
SEAVER RoBERTa w/ 3*orig.	372M	39.35	39.28
SimCSE (Deshpande et al., 2023)	125M	38.56	39.00
SEAVER SimCSE w/o orig.	132M	37.16	36.92
SEAVER SimCSE w/ 1*orig.	132M	38.48	38.08
SEAVER SimCSE w/ 2*orig.	**132M**	**39.59**	**39.30**
SEAVER SimCSE w/ 3*orig.	132M	39.18	39.24
SimCSE (Deshpande et al., 2023)	355M	42.28	42.40
SEAVER SimCSE w/o orig.	372M	43.06	43.01
SEAVER SimCSE w/ 1*orig.	372M	42.47	42.52
SEAVER SimCSE w/ 2*orig.	372M	43.70	43.47
SEAVER SimCSE w/ 3*orig.	**372M**	**43.83**	**43.81**

Table 7: Fine-tuning results in Spearman and Pearson correlation (scaled by 100) on the C-STS-2023 test set. Bold rows indicate the highest performance achieved within the same model and scale. "SEAVER [`MODEL NAME`] w/ α*orig." denotes the addition of the Self-Augmentation signal to the original output (scaled by a factor of α), and "w/o" is equivalent to $\alpha = 0$.

A.3 Model Parameter Discussion

In Table 3 and Table 7, we can observe that the parameter count of SEAVER has increased slightly

[3] https://huggingface.co/FacebookAI/roberta-base

[4] https://huggingface.co/FacebookAI/roberta-large

[5] https://huggingface.co/princeton-nlp/sup-simcse-roberta-base

[6] https://huggingface.co/princeton-nlp/sup-simcse-roberta-large

[7] https://huggingface.co/deepmind/language-perceiver

[8] https://github.com/bfshi/AbSViT

compared to the similar scale baseline, the application of our method results in an increase of 7M training parameters for `base` scale models and 17M for `large` scale models. This translates to our proposed method introducing **1.056** and **1.047** times the number of parameters of the fine-tuning baseline language model for `base` and `large` scales, respectively. This increase is due to the application of a projection matrix that maps the concatenated multi-head vector dimensions back to the model dimension (the slight increase in parameters corresponds to the introduction of this projection matrix).

However, since no external auxiliary information is introduced and the transformation is applied only to the information originally extracted by the model, our proposed SEAVER still maintains a relatively high degree of consistency with the original baseline model. And the increase in parameter count due to our approach has a negligible impact on training time and resource consumption. This consistency makes integrating our method into practice exceptionally efficient and convenient, eliminating the need for significant alterations to the existing structures and training methodologies of pre-trained language models.

As a supplement to the main body, in Table 7, we set the range of the scaling factor α in Eq. 6 from 0 to 3, to observe the impact on the overall model performance under different ratios of the Self-Augmentation signal combined with the original output.

As RoBERTa has not been fine-tuned on other STS datasets, it largely retains the general-purposed feature extraction capability acquired during pre-training. Therefore, for RoBERTa-base (125M), solely using the Self-Augmentation signal for prediction (i.e., setting α to 0) can yield its optimal result. Introducing varying degrees of the original output may, to some extent, impair this, leading to suboptimal performance. Conversely, the RoBERTa-large (355M), compared to RoBERTa-base, further enhances its feature extraction ability. With the increased depth of extracted features, some features suppressed in the Self-Augmentation signal can positively influence the prediction (due to increased learned semantic complexity; intuitively, some features may appear condition-irrelevant individually but become condition-relevant in combination), thus introducing a certain degree of the original output (i.e., setting α to 2) can achieve its optimal result.

While SimCSE has already been fine-tuned on unconditional STS datasets, we believe this slightly impairs the model's ability to extract general features. However, SimCSE also acquires effective task-specific features for measuring sentence similarity. There exists a certain trade-off between the negative and positive impacts brought by fine-tuning on the unconditional STS datasets. Intuitively, we suspect this is related to the model's scale. The SimCSE-base (125M) is more likely to be negatively influenced by fine-tuning on the unconditional STS datasets compared to SimCSE-large (255M), resulting in the optimal performance of SimCSE-base being lower than that of the same scaled RoBERTa. In contrast, SimCSE-large seems to gain more positive benefits than negative impacts from the unconditional STS fine-tuning process, thereby further enhancing its capability to extract semantic features and achieving higher optimal performance.

A.4 Self-Augmentation Ratio Analysis

We provide a more detailed trend analysis in this section. As shown in Figure 5, both the `base` and `large` scales of the RoBERTa model exhibited similar trends: a significant decrease in performance upon the initial introduction of the original output, followed by a pattern of first increasing and then continuing to decrease as α increases.

However, a distinction between the `base` and `large` scales of the RoBERTa model is observed in the performance peak upon increasing the degree of the original output's inclusion: the `large` scale of RoBERTa surpasses the performance of using solely the Self-Augmentation signal for prediction, whereas the `base` scale does not.

The `base` scale SimCSE model shows a trend where performance continuously grows to a peak and then declines as α increases. The performance trend of the `large` scale SimCSE model is similar to that of RoBERTa, but the peak performance appears to be shifted to the right. It is also observable that at this point, the performance improvement has begun to converge.

We can also observe from Figure 5 that the best configuration of α varies across models of different scales and training methodologies. This variation is due to differences in the intrinsic sentence understanding capabilities and preferences of each model. Models with weaker sentence understanding, such

as RoBERTa (pre-training + C-STS fine-tuning), typically require a smaller α value compared to SimCSE (pre-training + STS fine-tuning + C-STS fine-tuning) when both models are of the same scale. This indicates a greater need for a higher proportion of Self-Reweighting output, which serves primarily as a supplementary and modulatory signal, to facilitate attention reallocation. Models with higher intrinsic sentence comprehension have less need for attention reallocation through the Self-Reweighting output to mitigate distraction.

However, it is important to emphasize that the role of Self-Reweighting output in facilitating attention reallocation is still crucial even in models with stronger sentence understanding capabilities. This is evident as the model performance degrades to that of the corresponding baseline models when $\alpha \to +\infty$.

Sentence 1	Sentence 2	Condition	Output
Two martial artists compete before a referee and onlookers.	Two people are fighting in full protective gear and helmets.	The number of participants.	w/o: 2.90 w/ : 4.61 Label: 5.00
A man in a black wetsuit rides a surfboard on a wave.	Surfer in black wetsuit falling off his board into the water.	The color of clothing.	w/o: 2.75 w/ : 4.75 Label: 5.00
A man dressed in red dives for a shuttlecock with a racket on a court.	A Japanese man in a red shirt, at the olympics playing tennis.	The name of the color.	w/o: 2.35 w/ : 4.08 Label: 5.00
At a rodeo and a cowboy is riding a bull and other men are standing by.	A man dressed as a cowboy walks away from a brown horse.	The type of animals.	w/o: 3.35 w/ : 1.54 Label: 1.00
A youth on a skateboard is doing flips and tricks over a metal bar.	Young kid in a blue shirt is doing a trick on his rollerblades.	What the person is wearing on their feet.	w/o: 3.07 w/ : 1.28 Label: 1.00
A man with a blue harness climbing a climbing wall.	A young girl wearing a safety harness climbs a rock wall.	The sex of the person.	w/o: 3.37 w/ : 1.66 Label: 1.00
A guy in red shirt is rock-clibbing on a dangerous mountain wall.	A man in a red jacket mountain climbing an icy rock mountain.	The color of clothing.	w/o: 2.18 w/ : 4.12 Label: 5.00
A brown and white dog running fast in a fenced yard.	A dog is running while catching a tennis ball in its mouth.	The action.	w/o: 2.73 w/ : 4.47 Label: 5.00
A boy wearing a green shirt rides a scooter down the sidewalk.	A little boy in a green jacket is crying on his tricycle.	The color of the clothing.	w/o: 2.25 w/ : 4.10 Label: 5.00
A woman in an oversized black shirt plays a black and red guitar in a musky room.	A bass player girl, who is performing at a concert one of the bands songs.	The sex of the musician.	w/o: 2.58 w/ : 4.20 Label: 5.00

Table 8: 10 additional cases from the C-STS-2023 validation set. "Output" refers to the predicted and the ground-truth similarity, where the notation "w/o" represents the prediction from the baseline model, and "w/" denotes the prediction from our proposed SEAVER (based on RoBERTa-base).

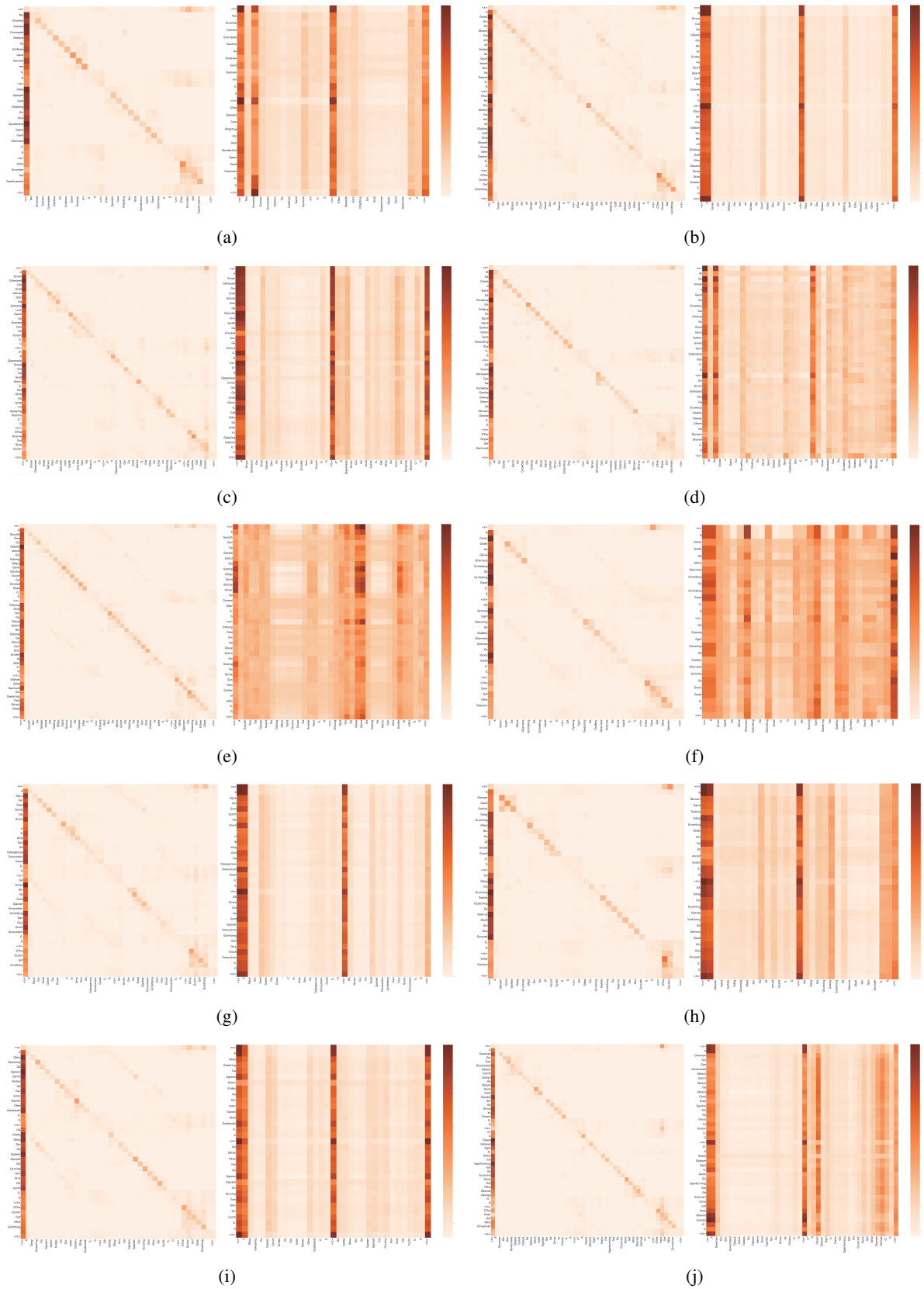

Figure 6: Average attention matrix (left: obtained from the baseline model) and Self-Reweighting weight (right: obtained from our proposed SEAVER) of each row case ((a) for the first row, (b) for the second row, etc) presented in Table 8. The darker the color, the larger the corresponding value.

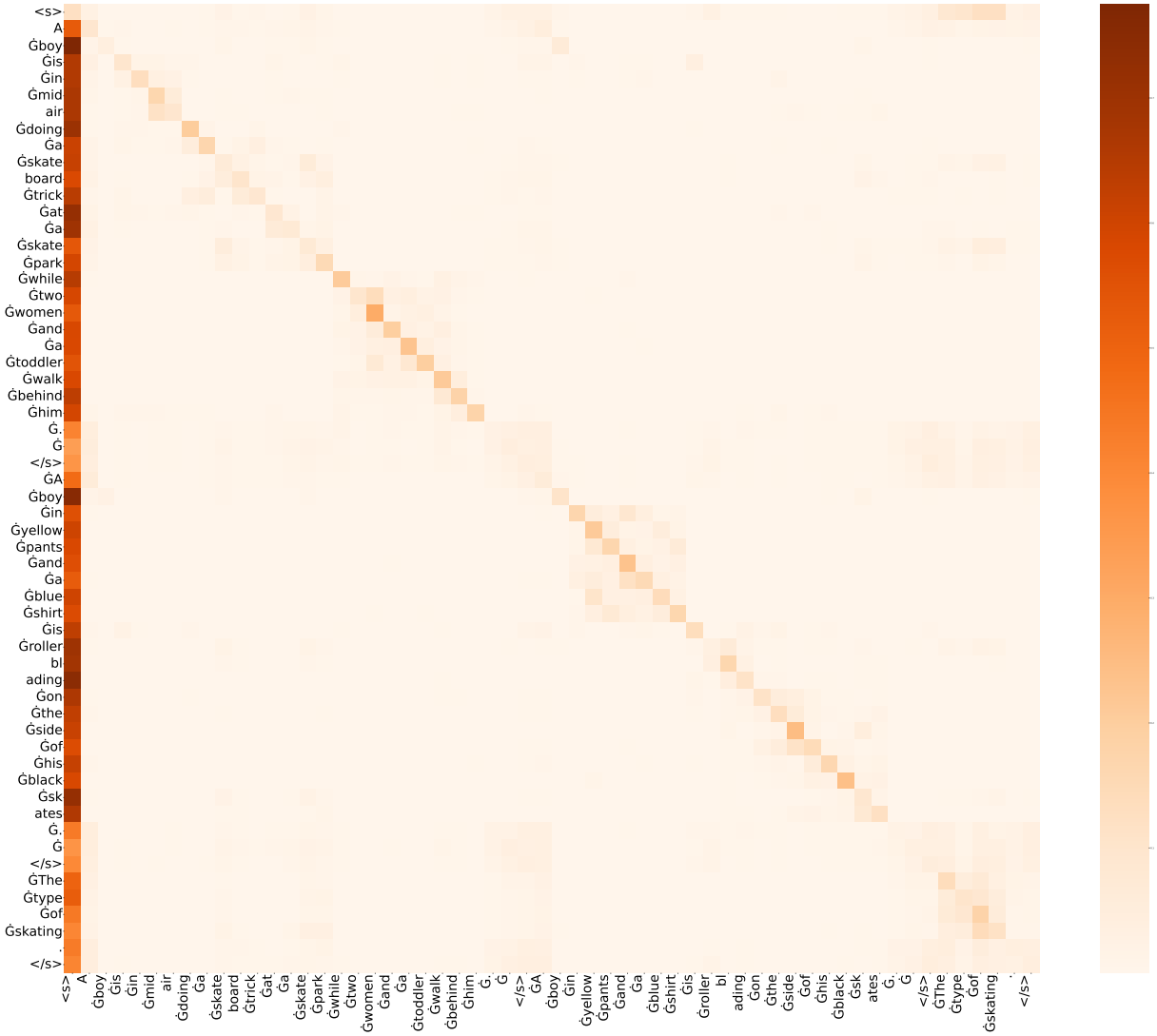

Figure 7: An enlarged version of Figure 4 (left), which is provided for a clearer display of tokens and attention details.

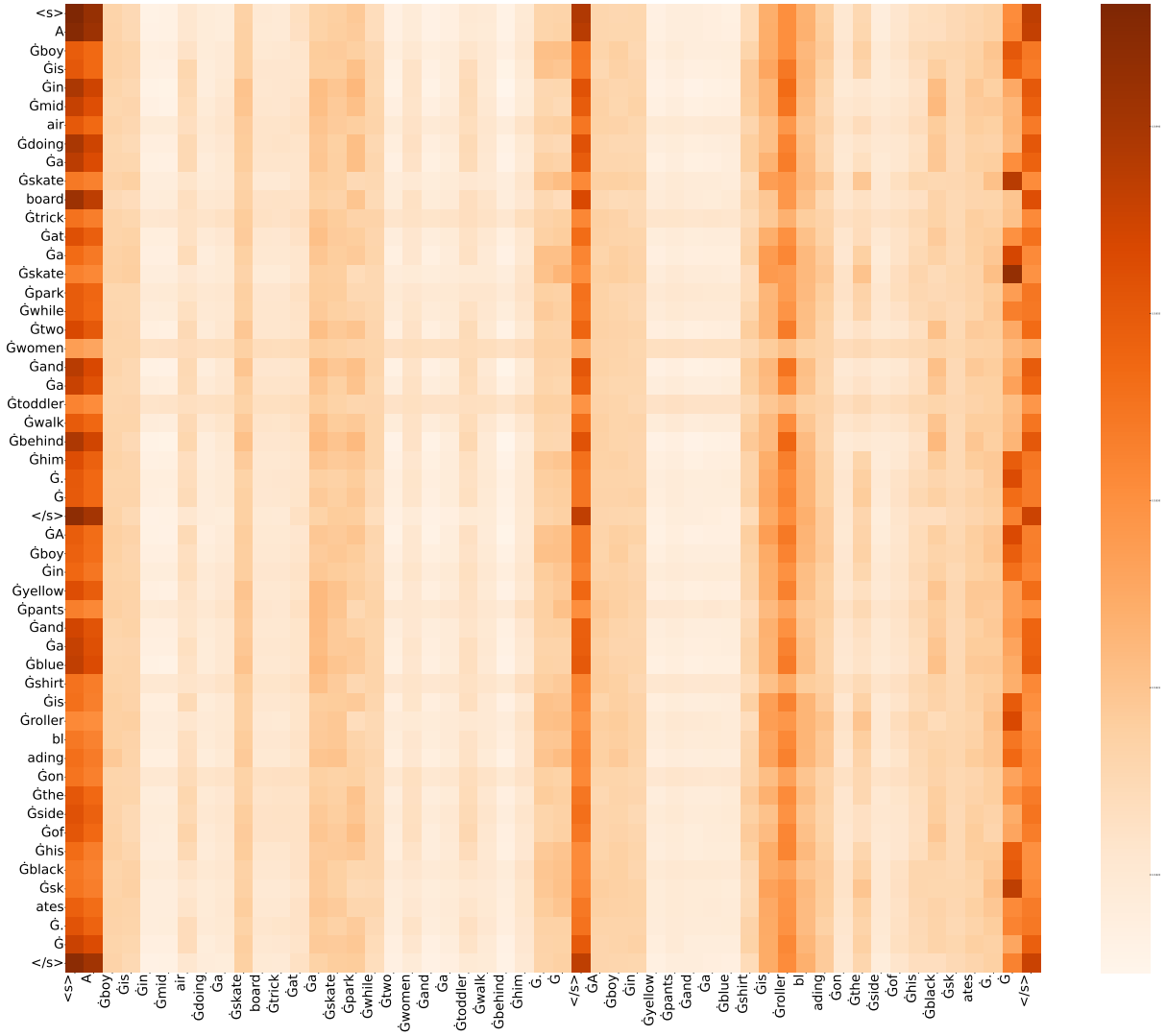

Figure 8: An enlarged version of Figure 4 (right), which is provided for a clearer display of tokens and attention details.