AoE: Angle-optimized Embeddings for Semantic Textual Similarity [*]

Xianming Li [1], **Jing Li** [1,2†]
[1] Department of Computing
[2] Research Centre on Data Science & Artificial Intelligence
The Hong Kong Polytechnic University
`xianming.li@connect.polyu.hk`, `jing-amelia.li@polyu.edu.hk`

Abstract

Text embedding is pivotal in semantic textual similarity (STS) tasks, which are crucial components in Large Language Model (LLM) applications. STS learning largely relies on the cosine function as the optimization objective to reflect semantic similarity. However, the cosine has saturation zones rendering vanishing gradients and hindering learning subtle semantic differences in text embeddings. To address this issue, we propose a novel Angle-optimized Embedding model, AoE. It optimizes angle differences in complex space to explore similarity in saturation zones better. To set up a comprehensive evaluation, we experimented with existing short-text STS, our newly collected long-text STS, and downstream task datasets. Extensive experimental results on STS and MTEB benchmarks show that AoE significantly outperforms popular text embedding models neglecting cosine saturation zones. It highlights that AoE can produce high-quality text embeddings and broadly benefit downstream tasks. The code is available at: `https://github.com/SeanLee97/AnglE`

1 Introduction

Text embeddings, essential language features, are foundations of semantic textual similarity (STS) tasks, which quantify how similar two text pieces are in semantics (Kiros et al., 2015; Hill et al., 2016; Conneau et al., 2017; Cer et al., 2018; Reimers and Gurevych, 2019; Gao et al., 2021). They broadly benefit downstream tasks, such as information retrieval (Asai et al., 2023) and clustering (Xu et al., 2023), and are particularly helpful in many recent LLMs-based applications (OpenAI, 2022a; Touvron et al., 2023); e.g., many RAG tasks employ text embeddings for retrieval (Asai et al., 2023).

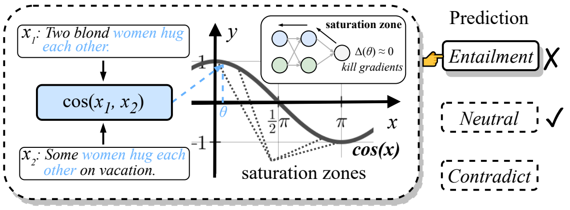


Figure 1: A case from SNLI. SimCSE and SBERT ignore the effects of cosine saturation zones and wrongly predict "*entailment*" due to shallow features of highly overlapping words, while the correct label is "*neutral*."

The existing STS training commonly involves optimizing cosine functions — the learning objective to indicate the similarity of pairwise text embeddings (Reimers and Gurevych, 2019; Gao et al., 2021; Su, 2022; Zhuo et al., 2023). However, the cosine has *saturation zones*, resulting in *gradient vanishing* in optimization regardless of the network depth (Roodschild et al., 2020). The gradient will be close to zero for embedding pairs falling in the saturation zone, preventing parameters from updating in backpropagation. Because embedding pairs in saturation zones are nearly aligned or anti-aligned, it hinders text embedding models from discerning subtle, implicit differences that appear similar yet are actually dissimilar in semantics.

Such pairs commonly appear in STS training data from Natural Language Inference (NLI) datasets, such as the Multi-Genre NLI (MNLI) (Williams et al., 2018) and the Stanford NLI (SNLI) (Bowman et al., 2015). They typically include three labels of *entailment*, *neutral*, and *contradict*; pairs in saturation zones may render obscure cross-label boundaries. To illustrate this point, Figure 1 shows an example from the SNLI dataset. The "*neutral*" pair shows a high appearance similarity (with many shared words) instead of semantically similar. The similar appearance similarity results in them falling into cosine's saturation zones, causing vanishing gradients during optimization. Consequently, the model mistakenly considers their relations as "*entailment*" instead of their correct label "*neutral*."

[*] Previously known as "AnglE"
[†] Corresponding author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1825–1839
August 11-16, 2024 ©2024 Association for Computational Linguistics

Viewing these concerns, we aim to tackle the negative effects of the cosine's saturation zones in embeddings and propose a novel Angle-optimized Embedding (AoE) model for STS. It decomposes an embedding into real and imaginary components through complex division, aiming to employ the real component for reflecting appearance differences and the imaginary component for subtle differences. It allows AoE to involve the optimization of the angle difference to understand subtle differences in text pairs for similarity learning.

To the best of our knowledge, *we are the first to explore the negative effects of cosine's saturation zones and optimize angle differences through division in complex space for text embedding learning.*

In the STS experimental setup, we observed that most existing STS benchmarks focus on evaluating models on short texts. Unfortunately, limited datasets are available to evaluate the STS performance on long texts. However, long texts are prevalent in real-world applications such as financial and legal documents (Li et al., 2023). To tackle this challenge, we present a high-quality long-text STS dataset collected from GitHub Issues with roughly 22K samples. It allows for a more comprehensive evaluation of STS performance with long texts.

We first experimented with short- and long-text STS datasets in the standard and in-domain STS tasks, where AoE outperforms non-trivial baselines in varying embedding backbones. Then, AoE shows consistently superior results in facilitating various downstream tasks, indicating its benefits in diverse scenarios. In particular, AoE achieves SOTA results on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022) at the same model scale. Next, an ablation study indicates that all modules positively contribute to AoE. Finally, we further discuss how AoE learns better embeddings in cosine saturation zones.

In summary, our contributions are as follows:

- We investigate the effects of cosine saturation zones for STS and optimize angle differences in complex space for improving text embedding.

- We extend the existing STS benchmark with a new long-text dataset from Github Issues to allow more comprehensive STS empirical studies.

- We present extensive experiments demonstrating that AoE effectively handles cosine saturation zones to broadly benefit text embedding learning and create positive effects in various scenarios.

2 Related Work

Our work is related to text embedding learning. Compared to early efforts focusing on word embeddings (Mikolov et al., 2013), text embeddings (a more general concept) enable semantic representation for richer context. Many prior studies (Li et al., 2020; Su et al., 2021a) employed pretrained models such as BERT (Devlin et al., 2019) for learning text embeddings without fine-tuning. More recent work showed the benefits of fine-tuning the pretrained models to improve STS. Some studies (Conneau et al., 2017; Cer et al., 2018) involved human-labeled training data, e.g., NLI datasets. In training methods, SBERT (Reimers and Gurevych, 2019) used siamese BERT, while many others adopted contrastive learning (Zhang et al., 2020; Gao et al., 2021; Chuang et al., 2022; Zhuo et al., 2023).

Most widely-used models employ cosine to measure similarity in their learning objectives, as shown in Table 6, Appendix A. Cosine exhibits saturation zones leading to gradient vanishing (Roodschild et al., 2020). It hinders embedding models from encoding subtle differences in pairs falling in the saturation zones. However, none of the existing work considers such an issue, and we propose the angle-optimized AoE model to mitigate this gap.

AoE is inspired by **complex embeddings** for using complex division to exploit angle differences. In positional word embedding learning, Su et al. (2021b) adopted complex multiplication to introduce rotary position embedding into transformer architecture. In knowledge graph learning, Trouillon et al. (2016); Sun et al. (2019) presented entity embedding in complex space to model the source to target rotations for link prediction. However, their embeddings are *on the word or entity level* and focus on their position modeling. On the contrary, we make the first efforts to leverage complex division to compute normalized angle differences between *sentence-level* text embeddings for mitigating the negative effects of the cosine's saturation zones.

3 AoE Framework

Here, we will introduce the AoE methods with the overall framework in Figure 2. We will first present how to encode the complex text embeddings in Section 3.1, followed by the angle objective in Section 3.2. At last, Section 3.3 will outline the final learning objective to describe AoE's training process.

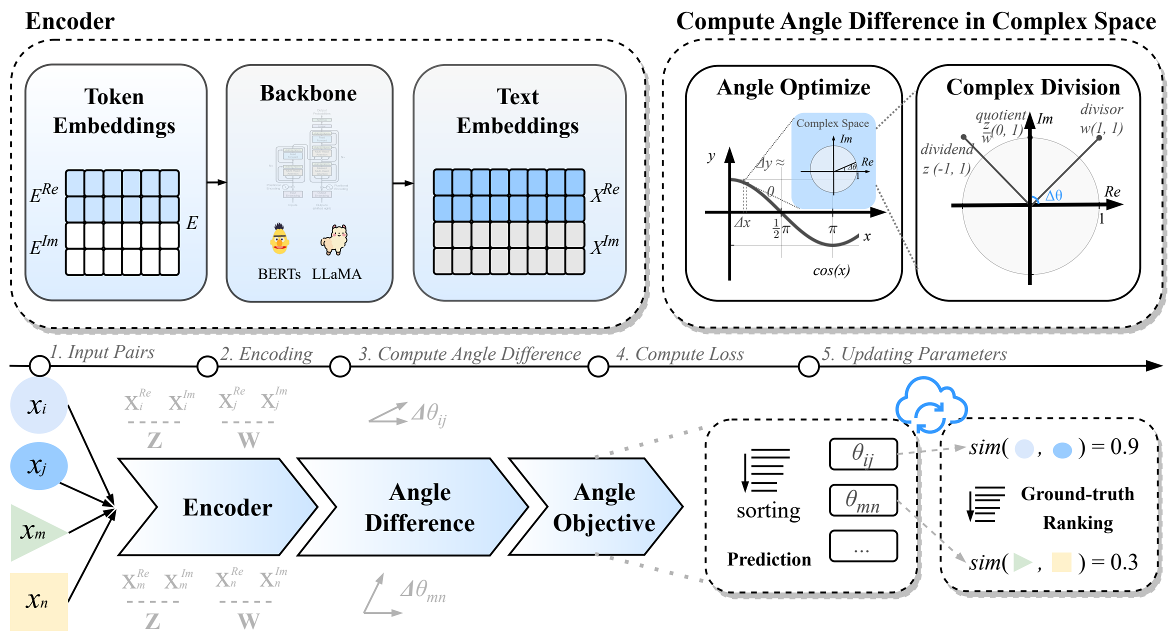


Figure 2: The overall framework of AoE. Initially, the input text pairs (x_i, x_j) and (x_m, x_n) are processed by the encoder to obtain real and imaginary text embeddings: ($\mathbf{X}_i^{re}$, $\utilde{\mathbf{X}_i^{im}}$), ($\mathbf{X}_j^{re}$, $\utilde{\mathbf{X}_j^{im}}$), ($\mathbf{X}_m^{re}$, $\utilde{\mathbf{X}_m^{im}}$), and ($\mathbf{X}_n^{re}$, $\utilde{\mathbf{X}_n^{im}}$). After obtaining these complex text embeddings, the angle differences, $\Delta\theta_{ij}$ and $\Delta\theta_{mn}$, can be computed. Finally, the angle differences are then used in the optimization of the angle objective.

3.1 Complex Text Embeddings

The first step of AoE is to transform the input text into complex text embeddings. Our intuition is to employ real components for learning appearance differences (like prior practices (Reimers and Gurevych, 2019; Gao et al., 2021)) and imaginary ones for subtle semantic differences. This way, we can exploit the angle differences for embeddings' similarity learning in cosine saturation zones.

To achieve this, we first input the text x into the embedding layer to obtain the token embeddings: $\mathbf{E} = \mathrm{Emb}(x) \in \mathbb{R}^{2d}$. Inspired by Sun et al. (2019), the token embeddings $\mathbf{E}$ include two sub-spaces. The first d embeddings represent the real token embeddings $\mathbf{E}^{re} = \mathbf{E}_{1:d} \in \mathbb{R}^d$, while the embeddings from d to $2d$ represent the imaginary token embeddings, denoted as $\mathbf{E}^{im} = \mathbf{E}_{d:2d} \in \mathbb{R}^d$.

Then, the token embeddings are fed into Transformer encoders like BERT (Devlin et al., 2019) or LLaMA (Touvron et al., 2023) to obtain the text embeddings: $\mathbf{X} = \mathrm{Encoder}_p(\mathbf{E}) \in \mathbb{R}^{2d}$, where p means pooling. Specifically, embeddings of the "CLS" token for BERT and the last token for LLaMA represent sentence-level text embeddings. Consequently, a text embedding has real ($\mathbf{X}^{re} = \mathbf{X}_{1:d} \in \mathbb{R}^d$) and imaginary components ($\utilde{\mathbf{X}^{im}} = \mathbf{X}_{d:2d} \in \mathbb{R}^d$), where $\utilde{wave}$ henceforth indicates the imaginary embeddings for easy reading.

3.2 Angle Objective

After obtaining the complex text embeddings, we present the angle objective in complex space. Specifically, for the input text pair (x_i, x_j), we obtain their real and imaginary text embeddings ($\mathbf{X}_i^{re}$, $\utilde{\mathbf{X}_i^{im}}$) and ($\mathbf{X}_j^{re}$, $\utilde{\mathbf{X}_j^{im}}$) via the process in Section 3.1. To allow clearer formula derivation, we define:

$$\mathbf{z} = \mathbf{a} + \utilde{\mathbf{b}}i \in \mathbb{C}, \mathbf{w} = \mathbf{c} + \utilde{\mathbf{d}}i \in \mathbb{C}, \tag{1}$$

where $\mathbf{a} = \mathbf{X}_i^{re} \in \mathbb{R}^d$, $\utilde{\mathbf{b}} = \utilde{\mathbf{X}_i^{im}} \in \mathbb{R}^d$, $\mathbf{c} = \mathbf{X}_j^{re} \in \mathbb{R}^d$, and $\utilde{\mathbf{d}} = \utilde{\mathbf{X}_j^{im}} \in \mathbb{R}^d$. Then, we conduct complex division to determine the angle difference and the factor in magnitude. Based on the complex division rule, we can measure the angle difference between embeddings $\mathbf{z}$ and $\mathbf{w}$, $\Delta\theta_{zw}$, as follows:

$$\Delta\theta_{zw} = \log\left[\frac{(\mathbf{ac} + \utilde{\mathbf{bd}}) + (\utilde{\mathbf{b}}\mathbf{c} - \mathbf{a}\utilde{\mathbf{d}})}{\sqrt{(\mathbf{c}^2 + \utilde{\mathbf{d}}^2)(\mathbf{a}^2 + \utilde{\mathbf{b}}^2)}}\right], \tag{2}$$

where the denominator serves as the normalization term (naturally derived from complex division). For detailed derivation, we refer readers to Appendix C. Based on that and following Su (2022), we optimize the angle difference between input text pairs with the ranking objective function below:

$$\mathcal{L}_{angle} = \log \left[1 + \sum_{s_{ij} > s_{mn}} \exp(\frac{\Delta\theta_{ij} - \Delta\theta_{mn}}{\tau}) \right], \quad (3)$$

where τ is a temperature hyperparameter. s_{ij} is the similarity between text x_i and x_j, and s_{mn} is the similarity between text x_m and x_n. $s_{ij} > s_{mn}$ is from the ranking of training data labels. By optimizing the angle objective, $\mathcal{L}_{angle}$, we aim to minimize the angle difference for pairs with high similarity compared to those with low similarity. Thus, for embedding pairs in cosine saturation zones (e.g., similar ones in appearance), the angle objective helps reflect the subtle semantic differences, mitigating the negative effects of gradient vanishing.

3.3 Training Process of AoE Framework

In the embedding training of AoE, we optimize the angle objective (Section 3.2) with the auxiliary objective. This multi-objective approach allows the AoE framework to learn text embeddings comprehensively from multiple perspectives, enhancing the model's overall performance. Here, we employ the widely-used supervised contrastive learning objective as the auxiliary objective $\mathcal{L}_{cl}$, as follows:

$$\mathcal{L}_{cl} = -\sum_{b} \sum_{i}^{m} \log \left[\frac{e^{\cos(\mathbf{X}_{b_i}, \mathbf{X}^{+}_{b_i})/\tau}}{\sum_{j}^{N} e^{\cos(\mathbf{X}_{b_i}, \mathbf{X}^{+}_{b_j})/\tau}} \right], \quad (4)$$

where τ is a temperature hyperparameter, b stands for the b-th batch, $\mathbf{X}^{+}_{b_i}$ and $\mathbf{X}^{+}_{b_j}$ are the respective positive samples of $\mathbf{X}_{b_i}$ and $\mathbf{X}_{b_j}$, m represents the number of positive pairs in b-th batch, N is the batch size, and $\cos(\cdot)$ is the cosine similarity.

In the training, we combine the angle objective and the contrastive objective in the following manner to form the final objective function:

$$\mathcal{L} = w_1 \cdot \mathcal{L}_{angle} + w_2 \cdot \mathcal{L}_{cl}, \quad (5)$$

where w_1 and w_2 are two hyperparameters to control the weights of balancing the two objectives.

4 Experimental Setup

Here, we elaborate on the experimental setup, including datasets, baselines, evaluation metrics, and implementation details. We also open-source our trained models in Appendix Section E.

Datasets. Following standard setup (Gao et al., 2021), the training data is from MultiNLI and SNLI. Our statistics reveal that 33% of the text pairs show a similarity above 0.95 and 66% above 0.8. It means a large proportion of samples in or near cosine saturation zones, implying the challenges of learning subtle semantic differences for them.

For evaluation, we test AoE on STS tasks with existing widely-used short-text STS datasets and our newly proposed long-text GitHub Issue Similarity Dataset. Furthermore, we examine AoE on downstream data with 7 popular tasks and MTEB.

• *Existing Short-text STS Tasks.* We first evaluate AoE on 7 widely-adopted STS datasets, namely: STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), SICK-R (Marelli et al., 2014), and STS-B (Cer et al., 2017). These datasets mainly consist of short text (less than 512 tokens), whereas real-world scenarios often involve long texts. Viewing this gap, we introduce a new long-text dataset called GitHub Issues Similarity Dataset (GIS) as follows for a more extensive STS evaluation.

• *GitHub Issues Similarity Dataset (GIS).* The GIS dataset was gathered based on the GitHub duplicate issues indicating high similarity. The duplication label is easy to access because maintainers of open source organizations tend to mark these duplicate issues as closed with a comment like "closing as a duplicate of #id." Consequently, these duplicate issues inherently serve as a source of the STS task. Here, most issues contain long text because of the large amount of code involved.

We extracted duplicated issues from 55 famous open-source projects (see Appendix B) on GitHub using GitHub API to compile the dataset. The duplicate issues served as positive samples, while the remaining ones were considered negative. These open-source projects have active participation from maintainers and volunteers to maintain the issue quality. Additionally, We randomly selected 10% of the data for manual inspection, and the quality was found to be satisfactory. 93% of the sampled data can be clearly classified as either similar with label 0 or dissimilar with label 1. Our statistics show that the proportion of long text (with token length > 512) for the train, validation, and test sets is 61.03%, 60.85%, and 60.50%, respectively. More details of GIS are presented in Appendix B.[1]

• *Downstream Tasks.* Following standard practice, we evaluate AoE on 7 downstream tasks: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe

[1] The dataset can be downloaded at https://hf.co/datasets/WhereIsAI/github-issue-similarity.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Closed-source Models								
openai-ada-002	69.80	83.27	76.09	86.12	85.96	83.17	80.60	80.72
openai-text-embedding-3	72.84	86.10	81.15	88.49	85.08	83.56	79.00	82.32
Open-source Models								
InferSent-GloVe †	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
USE †	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
BERT$_{base}$								
ConSERT	74.07	83.93	77.05	83.66	78.76	81.36	76.77	79.37
CoSENT	71.35	77.52	75.05	79.68	76.05	78.99	71.19	75.69
SBERT †	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SimCSE	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
AoE (ours)	$75.26_{\pm 0.04}$	$85.61_{\pm 0.06}$	$80.64_{\pm 0.12}$	$86.36_{\pm 0.11}$	$82.51_{\pm 0.15}$	$85.64_{\pm 0.10}$	$80.99_{\pm 0.09}$	82.43
LLaMA$_{7B}$								
SBERT ⋆	$77.58_{\pm 0.15}$	$89.21_{\pm 0.31}$	$84.32_{\pm 0.33}$	$87.63_{\pm 0.28}$	$85.78_{\pm 0.40}$	$87.06_{\pm 0.31}$	$80.95_{\pm 0.29}$	84.65
SimCSE ⋆	$78.39_{\pm 0.12}$	$89.95_{\pm 0.23}$	$84.80_{\pm 0.19}$	$88.50_{\pm 0.40}$	$86.04_{\pm 0.29}$	$87.86_{\pm 0.35}$	$81.11_{\pm 0.43}$	85.24
AoE (ours)	$79.00_{\pm 0.12}$	$90.56_{\pm 0.21}$	$85.79_{\pm 0.18}$	$89.43_{\pm 0.36}$	$87.00_{\pm 0.29}$	$88.97_{\pm 0.32}$	$80.94_{\pm 0.29}$	85.96
LLaMA$_{13B}$								
SBERT ⋆	$78.03_{\pm 0.12}$	$89.89_{\pm 0.32}$	$85.03_{\pm 0.28}$	$88.96_{\pm 0.31}$	$86.12_{\pm 0.41}$	$88.03_{\pm 0.44}$	$81.11_{\pm 0.47}$	85.31
SimCSE ⋆	$78.69_{\pm 0.19}$	$90.58_{\pm 0.31}$	$85.50_{\pm 0.24}$	$89.56_{\pm 0.25}$	$86.92_{\pm 0.37}$	$88.92_{\pm 0.37}$	$81.28_{\pm 0.44}$	85.92
AoE (ours)	$\mathbf{79.33}_{\pm 0.18}$	$\mathbf{90.65}_{\pm 0.28}$	$\mathbf{86.89}_{\pm 0.21}$	$\mathbf{90.45}_{\pm 0.26}$	$\mathbf{87.32}_{\pm 0.33}$	$\mathbf{89.69}_{\pm 0.38}$	$\mathbf{81.32}_{\pm 0.42}$	**86.52**

Table 1: Text embedding performance on the standard STS tasks. The blue cell background indicates that our results are the best among the corresponding backbones. The results highlighted in bold represent the global best performance. Results † are obtained from (Reimers and Gurevych, 2019). Results ⋆ denote our implementation using the official code. For the remaining baselines, we obtain their results from their original papers. Given any backbone, the paired t-test reveals significant improvements in AoE compared to all baselines with p-values $< 5\%$.

Model	STS-B	GIS	Avg. Spearman's
SimCSE	$76.27_{\pm 0.23}$	$60.38_{\pm 0.18}$	68.33
SBERT	$84.67_{\pm 0.35}$	$69.50_{\pm 0.47}$	77.09
AoE	$\mathbf{86.28}_{\pm 0.19}$	$\mathbf{70.59}_{\pm 0.35}$	**78.44**

Table 2: Results of the in-domain STS tasks. All baselines are our implementation using the official code. BERT$_{base}$ is the backbone for all models.

et al., 2005), SST2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000), and MRPC (Dolan et al., 2004). These tasks mainly evaluate the classification performance of text embeddings. We also examine AoE on the **MTEB** (Muennighoff et al., 2022) for a more thorough downstream task evaluation. It includes classification (12 datasets), clustering (11 datasets), pair classification (3 datasets), reranking (4 datasets), retrieval (15 datasets), STS (10 datasets), and summarization (1 dataset) tasks.

Evaluation Metrics. For *STS*, we follow previous studies to use SentEval (Conneau and Kiela, 2018) to compute Spearman's correlation and report the "all" setting. For *downstream tasks*, we employ SentEval to assess the performance of text embeddings. For a fair comparison, we follow baselines and use the default parameters of SentEval. For *MTEB*, we employ the official MTEB evaluation code to test the performance of AoE.

For all our implementations, we will report the average score over five runs and the std value (±).

Baselines. Because AoE is supervised, we primarily compare it with widely used supervised embedding baselines for a fair comparison. They are: InferSent (Conneau et al., 2017), USE (Cer et al., 2018), SBERT (Reimers and Gurevych, 2019), CoSENT (Su, 2022), and supervised versions of SimCSE (Gao et al., 2021) and ConSERT (Yan et al., 2021). In particular, given different backbones, we compare AoE with SBERT and SimCSE, the two most widely-used text embedding baselines. All the above baselines are open-source embeddings. In addition, we adopt two popular closed-source baselines, OpenAI's Ada-002 (OpenAI, 2022b) and OpenAI's text-embeddings-3 (OpenAI, 2024), for a comprehensive comparison.

Implementation Details. We extensively examine AoE on three scales of pre-trained backbone models: BERT$_{base}$ (uncased) (Devlin et al., 2019), LLaMA$_{7B}$ (LLaMA2-7B) (Touvron et al., 2023) and its counterpart in 13B. As for BERT, we set the initial learning rate to $5e-5$. For LLaMA, we apply the QLoRA (Dettmers et al., 2023) technique

Model	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	Avg.
openai-ada-002 ◇	–	–	–	–	–	–	–	90.10
Avg. BERT †	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
BERT-CLS †	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-BERT	81.09	87.18	94.96	88.75	85.96	88.64	74.24	85.83
DiffCSE-BERT$_{base}$	82.69	87.23	95.23	89.28	86.60	90.40	76.58	86.86
SimCSE-BERT$_{base}$	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
SBERT$_{base}\star$	80.10	86.25	94.61	88.78	84.90	89.00	73.25	85.27
AoE-BERT$_{base}$ (ours)	83.00$_{\pm 0.24}$	89.38$_{\pm 0.27}$	94.72$_{\pm 0.31}$	89.87$_{\pm 0.46}$	87.20$_{\pm 0.23}$	89.00$_{\pm 0.45}$	75.54$_{\pm 0.39}$	86.96
AoE-LLaMA$_{7B}$ (ours)	90.54$_{\pm 0.27}$	**93.06**$_{\pm 0.32}$	96.14$_{\pm 0.40}$	91.61$_{\pm 0.45}$	**95.00**$_{\pm 0.28}$	95.80$_{\pm 0.58}$	74.90$_{\pm 0.38}$	91.01
AoE-LLaMA$_{13B}$ (ours)	**90.77**$_{\pm 0.33}$	93.01$_{\pm 0.33}$	**96.15**$_{\pm 0.45}$	**91.83**$_{\pm 0.48}$	94.95$_{\pm 0.27}$	**96.60**$_{\pm 0.60}$	**76.87**$_{\pm 0.43}$	**91.45**

Table 3: Results of text embeddings on the downstream classification tasks. The reported metrics is accuracy. ◇: results from (OpenAI, 2022b); †: results from Reimers and Gurevych (2019); ⋆: results are our implementation using the official code. For the remaining baselines, we obtain their results from their original papers.

for efficient fine-tuning with the initial learning rate to $1e-4$. For embeddings, we used the prompt "*Summarize sentence {text} in one word:*" to obtain the summative token and concatenate it to the last token of the text and then apply its token embeddings as the text embeddings, inspired by (Jiang et al., 2023). For the temperature in objectives, we set the τ to 0.05 following the previous practice (Gao et al., 2021). For w_1 and w_2 in Equation 5, we use the grid search strategy to search for their values. For a fair comparison with prior work, we follow SimCSE (Gao et al., 2021) to set the random seed to 42 for all main experiments. Yet, in Section 5.2, we test AoE without fixed random seeds to examine its robustness.

5 Experimental Results

Section 5.1 first presents the main comparison results, followed by an ablation study in Section 5.2. Finally, we will further discuss AoE in Section 5.3.

5.1 Main Comparison Results

In the experimental comparison, we examine benchmark results of STS and downstream tasks for intrinsic and extrinsic embedding evaluations.

Standard STS. We begin with the standard STS benchmark experiments for models trained using MultiNLI and SNLI datasets and evaluated on SentEval. The results are presented in Table 1, where we can draw the following observations.

First, larger backbone models generally result in better performance. It implies the larger model scales of LLMs can helpfully capture deeper semantics for the STS prediction. Second, SimCSE works better than SBERT, possibly benefitting from contrastive learning for capturing semantic similarity. Third, given any backbone, AoE consistently performs best in all STS benchmarks. For example, compared to SimCSE, AoE demonstrates average score improvements of 0.86%, 0.72%, and 0.60% for BERT$_{base}$, LLaMA$_{7B}$, and LLaMA$_{13B}$, respectively. While sharing a contrastive learning objective with SimCSE, AoE's performance gain likely comes from the novel addition of the angle objective. It allows optimizing the angle differences to explore the subtle semantic differences of training samples in cosine saturation zones, which is prevalent in the training data as we showed in Section 4. Furthermore, AoE's improvements observed across different backbone model sizes indicate that the benefit from the angle objective is universal.

In-domain STS. To further examine the embedding training specifically, we experimented with in-domain STS tasks for STS-B with short text and GIS with long text. Here, the training and test sets are obtained from the same dataset, and BERT$_{base}$ is the backbone for efficiency restrictions with long text. Table 2 presents the results. As can be seen, all models perform much worse on GIS than STS-B. It implies that long-text STS presents non-trivial challenges, requiring more in-depth exploration. We also note that SimCSE performs worse than SBERT (opposite to standard STS). It indicates that contrastive learning may rely on large-scale training samples, which is inadequate for in-domain STS. Nevertheless, AoE consistently performs the best, achieving improvements of 1.35% and 10.11% compared to SBERT and SimCSE, respectively. It indicates that the angle objective may enable more efficient STS training, reducing reliance on large-scale training data.

Downstream Tasks. The above experiments concerned intrinsic evaluations. For extrinsic evaluations, we assess how the embeddings can benefit downstream tasks. Here, we first consider 7 popular classification tasks and show the results in Table 3. We can see that AoE-BERT$_{base}$ performs better than other BERT$_{base}$ baselines, showing the subtle semantics captured by the angle objective can further benefit downstream tasks. Moreover, AoE-LLaMA$_{13B}$ achieves the best performance. These results indicate that AoE can produce text embeddings that helpfully assist downstream tasks.

MTEB Benchmark. We have shown the superiority of AoE embeddings on classification. Here, the leaderboard experiments of MTEB benchmark further provide a more extensive study in downstream tasks (Muennighoff et al., 2022). We trained AoE using the widely-used embedding data[2] and the supervised data released by BGE (Zhang et al., 2023). In the experimental results, AoE achieved SOTA performance in BERT-large scale models, with an average score of 64.64. Specifically, AoE outperformed the top 2 open-source BERT-large models: *bge-large-en-v1.5* (64.23) and *ember-v1* (63.54). Moreover, it outperforms popular closed-source models: *openai-text-embedding-3-large* (OpenAI, 2024) (64.59), *voyage-lite-01-instruct* (64.49), and *Cohere-embed-english-v3.0* (64.47). Most aforementioned comparison models are based on contrastive learning. It indicates that our novel angle objective design can provide performance gain for more challenging downstream tasks.

5.2 Ablation Study

While AoE has demonstrated overall effectiveness, we conduct ablation studies on the standard STS to investigate the contributions of AoE's different modules. The results are shown in Table 4.

First, we examine AoE's performance with varying objectives. Interstingly, using only the angle objective outperforms the counterpart using only the contrastive objective. It indicates that our angle objective might be more effective in learning semantic similarity than the contrastive objective. Nevertheless, combining both of them yields the best results. Second, we test AoE's performance on four different pooling strategies and find that the "cls" pooling is the most helpful one. Third, we test how random seeds affect AoE, and the results show that AoE is not sensitive to random seeds and robustly effective across varying selections.

Model	Avg. Spearman's Correlation
Objective	
AoE	$\mathbf{82.43}_{\pm 0.08}$
only angle objective	$82.36_{\pm 0.14}$
only contrastive objective	$81.53_{\pm 0.19}$
Pooling Strategy	
cls	$\mathbf{82.43}_{\pm 0.11}$
avg	$81.69_{\pm 0.18}$
max	$77.96_{\pm 0.21}$
Random Seed	
fixed random seed=42	$82.43_{\pm 0.10}$
different random seeds	$\mathbf{82.45}_{\pm 1.42}$

Table 4: The ablation study of AoE on the standard STS benchmark with BERT$_{base}$. We report the average (Avg.) Spearman's correlation over varying datasets.

5.3 Further Discussions and Analyses

To provide more insight, we further probe into AoE's output to interpret why it enables effective embedding learning as follows. Besides, we discuss its efficiency (training time) in Appendix D.

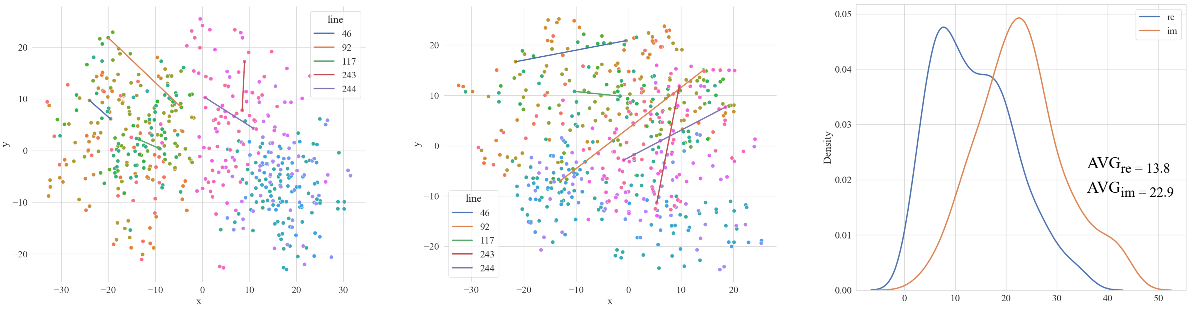
Model	MultiNLI	SNLI	Avg.
BERT$_{base}$	$52.56_{\pm 0.22}$	$61.646_{\pm 0.27}$	57.10
SimCSE-BERT$_{base}$	$54.57_{\pm 0.21}$	$62.38_{\pm 0.27}$	58.48
AoE-BERT$_{base}$	$\mathbf{56.60}_{\pm 0.19}$	$\mathbf{63.88}_{\pm 0.25}$	**60.24**

Table 5: Results on NLI tasks (by accuracy). All results are our implementation using the official code.

NLI Performance. Recall that we employed NLI datasets to train text embeddings, whereas the downstream task benchmarks do not involve NLI (see Table 3). We are hence interested in how AoE embeddings can benefit NLI tasks. Here, AoE is mainly compared with BERT and SimCSE text embeddings. Specifically, we input "*[CLS]premise[SEP]hypothesis[SEP]*" into the model and extract the representation of the "*[CLS]*" token for the logistic regression classification following SentEval (Conneau and Kiela, 2018). For the MultiNLI task, we report the average accuracy of *validation_matched* and *validation_mismatched* datasets. For the SNLI task, we report accuracy on the test set. The results are presented in Table 5. AoE consistently outperforms BERT and SimCSE. It suggests that AoE can well capture subtle semantics via training and thus benefit NLI tasks.

Real and Imaginary Text Embeddings in Cosine Saturation Zone. We then study what is encoded

[2] https://huggingface.co/embedding-data



(a) *Re* Text Embedding

(b) *Im* Text Embedding

(c) Distance Distribution

Figure 3: The t-SNE visualization of the real (*Re*) and imaginary (*Im*) text embeddings and the kernel density estimate plot of the real and imaginary distance between text pairs in the saturation zone of the STS-B test. AVG$_{re}$ and AVG$_{im}$ indicate the average distance between text pairs of the real and imaginary text embeddings.

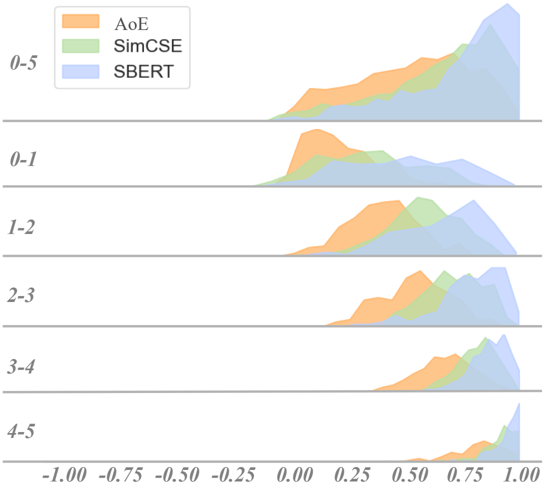


Figure 4: Density plots of cosine similarities between text pairs in the STS-B test set. The y-axis denotes the ground truth ratings (higher ratings indicate higher similarities). The x-axis is the cosine similarity.

in imaginary text embeddings to tackle text pairs in cosine saturation zones. To that end, we focus on the data in the STS-B test set's saturation zone (weighted similarity score > 0.95) and visualize them in a 2D plot using t-SNE (Van der Maaten and Hinton, 2008). Figures 3a and 3b show that imaginary text embeddings are more scattered than the real ones. To probe into the results, we draw lines between 5 sample text pairs and observe that the lengths of the lines, i.e., distances between text pairs, are larger for imaginary text embeddings than for real ones. For instance, consider one of the sample text pairs *Ukraine to implement unilateral ceasefire* and *Ukraine offers unilateral ceasefire*. The real distance is 11.8, while the imaginary distance is 22.1. This larger imaginary distance better reflects the subtle difference between "to implement" and "offers". AoE optimizes the angle differences to encode such subtle differences in the imaginary embeddings, resulting in better effectiveness. Figure 3c further supports this observation, as it shows that distances between text pairs are greater for imaginary text embeddings. This can be explained by the tendency that real text embeddings primarily capture appearance semantic differences, which can be influenced by saturation zones. Meanwhile, imaginary text embeddings specialize in capturing subtle semantic differences and help mitigate the negative effects of cosine saturation zones. We also visualize the full STS-B test set in Appendix D and have similar observations.

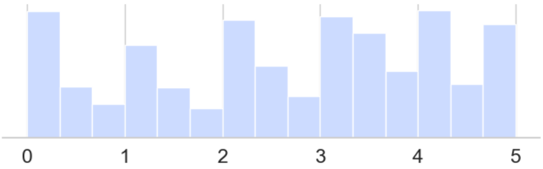


Figure 5: Density plots of golden (human annotated) scores between sentence pairs in the STS-B test set, ranging from 0 (dissimilar) to 5 (similar).

Text Embedding Distributions. Finally, we examine embedding distributions and how they align with the human senses. Figure 4 depicts the density plots of the cosine similarities between text pairs in the STS-B test set. Figure 5 shows the golden (human-labeled) scores, where human annotations are evenly distributed across varying similarity levels. However, Figure 4 implies that SimCSE and SBERT tend to focus their predictions within larger similarity intervals; in contrast, AoE's distribution leans towards the left, indicating its ability to utilize a broader range to diversify similarity predictions. It could be attributed to AoE's angle optimization, allowing imaginary embeddings to reflect subtle semantic differences. As a result, AoE's distribution aligns more closely with the humans' distribution.

6 Conclusion

In this paper, we have presented a novel text embedding model called AoE, which optimizes the angle difference in complex space to mitigate the negative effects of cosine saturation zones. To comprehensively evaluate AoE with STS tasks, we have introduced a GitHub Issues Similarity Dataset for long-text STS evaluation. Extensive experiments have suggested that AoE outperforms baselines, indicating that AoE can produce high-quality text embeddings and benefit various downstream tasks.

Ethics Statement

In this paper, we present a newly developed long-text STS dataset called GitHub Issue Similarity (GIS). The data collection process for GIS follows the guidelines of GitHub, and we use the official GitHub API to collect the necessary data. We have carefully reviewed the data and are confident that there are no ethical issues, such as offensive content. All repositories included in the GIS dataset are open source.

Limitations

One limitation of AoE lies in its performance improvement on our proposed long-text STS dataset GIS is comparatively lower than its performance on short-text STS tasks. We plan to improve AoE's performance on long-text STS tasks in future work.

Acknowledgements

This work is supported by the NSFC Young Scientists Fund (Project No. 62006203), a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU/25200821), the Innovation and Technology Fund (Project No. PRP/047/22FX), and PolyU Internal Fund from RC-DSAI (Project No. 1-CE1E).

Here, we sincerely thank the reviewers and ACs for their valuable input, which has greatly improved our work.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 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, Denver, Colorado. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

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

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46, Toronto, Canada. Association for Computational Linguistics.

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

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

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

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

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

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

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

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377. The Association for Computational Linguistics.

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

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

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3021–3035. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 3294–3302.

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

Xianming Li, Zongxi Li, Xiaotian Luo, Haoran Xie, Xing Lee, Yingbin Zhao, Fu Lee Wang, and Qing Li. 2023. Recurrent attention networks for long-text modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3006–3019, Toronto, Canada. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *27th Annual Conference on Neural Information Processing Systems 2013.*, pages 3111–3119.

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

OpenAI. 2022a. Introducing chatgpt.

OpenAI. 2022b. text-embedding-ada-002.

OpenAI. 2024. text-embedding-3.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

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*, pages 3980–3990. Association for Computational Linguistics.

Matías Roodschild, Jorge Gotay Sardiñas, and Adrián Will. 2020. A new approach for the vanishing gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, 9(4):351–360.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jianlin Su. 2022. Cosent (1): A more effective sentence vector scheme than sentence bert.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021a. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021b. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA. PMLR.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

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

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

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

Lingling Xu, Haoran Xie, Zongxi Li, Fu Lee Wang, Weiming Wang, and Qing Li. 2023. Contrastive learning models for sentence representations. *ACM Trans. Intell. Syst. Technol.*, 14(4).

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5065–5075. Association for Computational Linguistics.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*.

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

Wenjie Zhuo, Yifan Sun, Xiaohan Wang, Linchao Zhu, and Yi Yang. 2023. WhitenedCSE: Whitening-based contrastive learning of sentence embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12135–12148, Toronto, Canada. Association for Computational Linguistics.

A Related Work

Table 6 provides a list of models that include cosine similarity in their objective functions. We observe that using cosine similarity to measure similarity in objective functions that pose a gradient vanishing challenge is quite common.

Model	Use Cosine	Learning Algorithm
USE †	✗	Ensemble
SBERT ‡	✓	Regression
SimCSE ♣	✓	Contrastive Learning
ConSERT ♢	✓	Contrastive Learning
DiffCSE ♡	✓	Contrastive Learning
PromptCSE ♠	✓	Contrastive Learning
WhitenedCSE ♦	✓	Contrastive Learning

Table 6: The similarity measurements and learning algorithms of widely-used text embedding models. †: Cer et al. (2018). ‡: Reimers and Gurevych (2019). ♣: Gao et al. (2021). ♢: Yan et al. (2021). ♡: Chuang et al. (2022). ♠ is Jiang et al. (2022). ♦: Zhuo et al. (2023). The majority of them used cosine to measure similarity.

B Details of GIS Dataset

We collected GitHub issues via the official GitHub API from the following popular 55 repositories:

microsoft/terminal
mwaskom/seaborn
google/jax
twbs/bootstrap
JuliaLang/julia
microsoft/vscode
apache/airflow
electron/electron
apache/druid
apache/dubbo
scipy/scipy
scrapy/scrapy
babel/babel
vercel/next.js
golang/go
tiangolo/fastapi
webpack/webpack
neo4j/neo4j
facebook/react
bumptech/glide
keras-team/keras
mrdoob/three.js
celery/celery
rust-lang/rust
mui/material-ui
opencv/opencv
vuejs/vue
atom/atom
axios/axios
freeCodeCamp/freeCodeCamp
apache/shardingsphere
numpy/numpy
microsoft/playwright
scikit-learn/scikit-learn
apache/superset
denoland/deno
microsoft/PowerToys
kubernetes/kubernetes
symfony/symfony
flutter/flutter
microsoft/TypeScript
ansible/ansible
spring-projects/spring-framework
pandas-dev/pandas
angular/angular
elastic/elasticsearch
psf/requests
pytorch/pytorch
npm/cli
tensorflow/tensorflow
DefinitelyTyped/DefinitelyTyped
sqlalchemy/sqlalchemy
pallets/flask
huggingface/transformers
matplotlib/matplotlib

Figure 6 shows an example of the proposed GIS dataset. We can see that the texts are long, and there is a higher overlap among duplicate issues than non-duplicate issues.

Figure 7 shows the data source count distribution of the proposed GIS. We can observe that there is a wide range of repositories in GIS, most of which consist of over 100 samples.

Table 7 presents the data split and data size of the proposed GIS dataset, and Figure 9 depicts a violin plot illustrating the token-level text length distribution. The violin plot reveals a substantial number of lengthy texts.

Figure 8 depicts the distribution of the n-gram overlapping for non-duplicate and duplicate issue pairs. We can see that the overlapping becomes more significant as the grams decrease. Additionally, the overlapping of duplicate issue pairs is slightly larger than non-duplicate pairs. Specifically, the average overlapping of non-duplicate pairs for 1-gram, 2-gram, and 3-gram is 22, 12, and 9, respectively. Similarly, the average overlapping of non-duplicate pairs for 1-gram, 2-gram, and 3-gram is 26, 16, and 13, respectively. These statistics highlight the importance of using deep networks, even large language models, to identify duplicate and non-duplicate issues.

Split →	Train Set	Validation Set	Test Set
#Pos	9,457	774	807
#Neg	9,108	773	741
Total	18,565	1,547	1,548

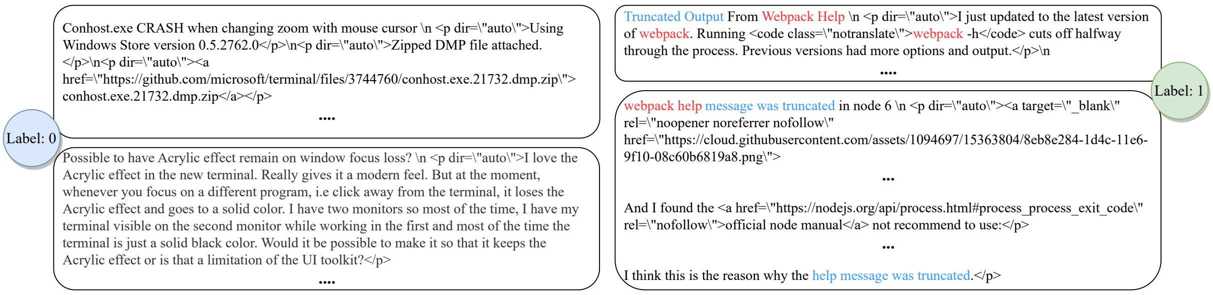
Table 7: Data split and data size of the GIS dataset. #Pos and #Neg is the count of positive and negative pairs, respectively.

C Detailed Derivation of Angle Difference

Complex division involves determining the angle difference and the factor in magnitude. Based on this, we calculate complex division between $\mathbf{z}$ and $\mathbf{w}$ as follows:

$$\begin{aligned} \frac{\mathbf{z}}{\mathbf{w}} &= \gamma e^{i\Delta\theta_{zw}} \\ \gamma &= \frac{r_{\mathbf{z}}}{r_{\mathbf{w}}} = \frac{\sqrt{\mathbf{a}^2 + \utilde{\mathbf{b}}^2}}{\sqrt{\mathbf{c}^2 + \utilde{\mathbf{d}}^2}} \\ \Delta\theta_{zw} &= \theta_{\mathbf{z}} - \theta_{\mathbf{w}}, \end{aligned} \tag{6}$$

where $r_{\mathbf{z}}$ and $r_{\mathbf{w}}$ represent the magnitudes of $\mathbf{z}$ and $\mathbf{w}$, while $\theta_{\mathbf{z}}$ and $\theta_{\mathbf{w}}$ denote the respective angles of



```
Conhost.exe CRASH when changing zoom with mouse cursor \n <p dir=\"auto\">Using
Windows Store version 0.5.2762.0</p>\n<p dir=\"auto\">Zipped DMP file attached.
</p>\n<p dir=\"auto\"><a
href=\"https://github.com/microsoft/terminal/files/3744760/conhost.exe.21732.dmp.zip\">
conhost.exe.21732.dmp.zip</a></p>
....
```

```
Possible to have Acrylic effect remain on window focus loss? \n <p dir=\"auto\">I love the
Acrylic effect in the new terminal. Really gives it a modern feel. But at the moment,
whenever you focus on a different program, i.e click away from the terminal, it loses the
Acrylic effect and goes to a solid color. I have two monitors so most of the time, I have my
terminal visible on the second monitor while working in the first and most of the time the
terminal is just a solid black color. Would it be possible to make it so that it keeps the
Acrylic effect or is that a limitation of the UI toolkit?</p>
....
```

```
Truncated Output From Webpack Help \n <p dir=\"auto\">I just updated to the latest version
of webpack. Running <code class=\"notranslate\">webpack -h</code> cuts off halfway
through the process. Previous versions had more options and output.</p>\n
....
```

```
webpack help message was truncated in node 6 \n <p dir=\"auto\"><a target=\"_blank\"
rel=\"noopener noreferrer nofollow\"
href=\"https://cloud.githubusercontent.com/assets/1094697/15363804/8eb8e284-1d4c-11e6-
9f10-08c60b6819a8.png\">

...

And I found the <a href=\"https://nodejs.org/api/process.html#process_process_exit_code\"
rel=\"nofollow\">official node manual</a> not recommend to use:</p>

...

I think this is the reason why the help message was truncated.</p>
```

Figure 6: An example of the proposed GIS dataset. The blue circle denotes non-duplicate issues labeled as 0, while the green one is duplicate issues labeled as 1. The "..." indicates the truncated text of the lengthy attached code.

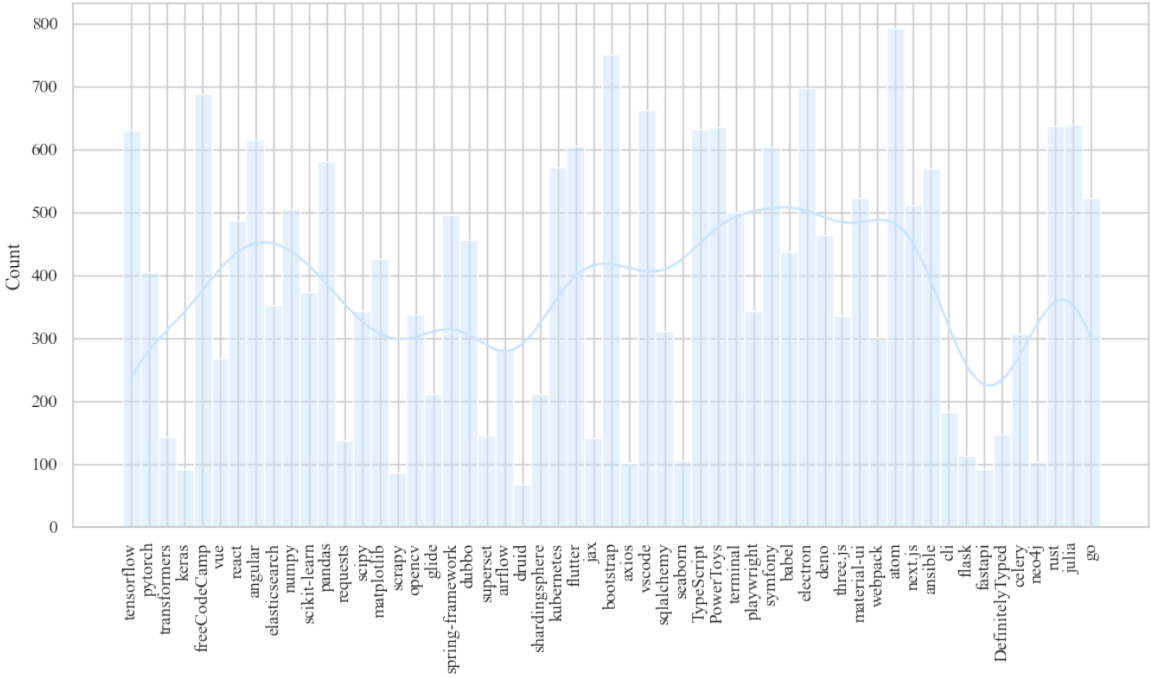


Figure 7: The distribution of data source counts in the proposed GIS dataset. The x-axis denotes the selected repository from GitHub.

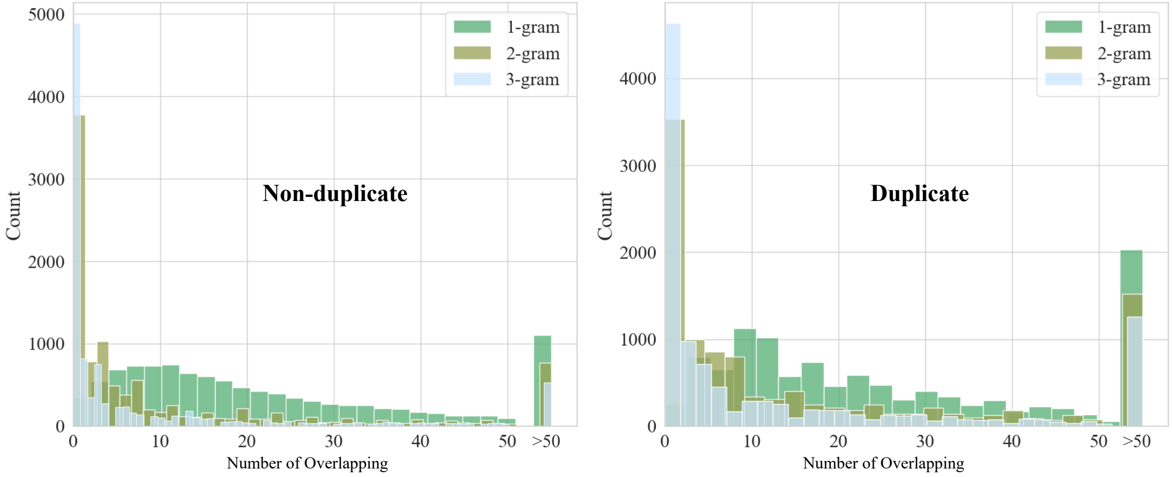


Figure 8: The distribution of the n-gram overlapping for the non-duplicate issue pairs and the duplicate issue pairs in the proposed GIS dataset.

URL	Description
Universal AoE Embedding Collection [a]	
https://hf.co/WhereIsAI/UAE-Large-V1 https://hf.co/WhereIsAI/UAE-Code-Large-V1	Universal AoE Embedding (English). AoE Embedding For Code Similarity
AoE NLI Embedding Collection [b]	
https://hf.co/SeanLee97/angle-bert-base-uncased-nli-en-v1 https://hf.co/SeanLee97/angle-llama-7b-nli-v2 https://hf.co/SeanLee97/angle-llama-13b-nli	BERT$_{base}$ NLI LLaMA2-7B NLI LLaMA2-13B NLI

Table 8: Pretrained models of AoE on HuggingFace.

[a] https://huggingface.co/collections/WhereIsAI/universal-angle-embeddings-663b0618ade1a39663e48190

[b] https://huggingface.co/collections/SeanLee97/angle-nli-sentence-embeddings-6646de386099d0472c5e21c0

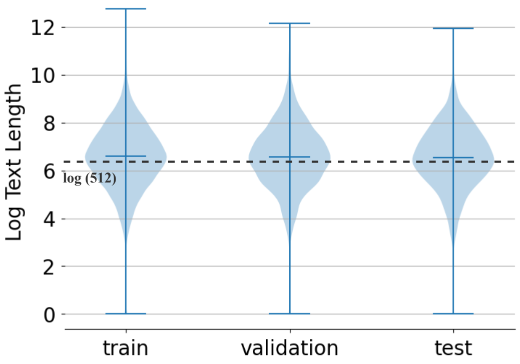


Figure 9: Log token-level length distribution of the GIS dataset. The red dashed line indicates the boundary line for long text. e servers as the base for log.

$\mathbf{z}$ and $\mathbf{w}$. Next, we compute the value of $\frac{\mathbf{z}}{\mathbf{w}}$ by the division rule in complex space as follows:

$$\frac{\mathbf{z}}{\mathbf{w}} = \frac{\mathbf{a} + \utilde{\mathbf{b}}i}{\mathbf{c} + \utilde{\mathbf{d}}i} = \frac{(\mathbf{ac} + \utilde{\mathbf{b}}\utilde{\mathbf{d}}) + (\utilde{\mathbf{b}}\mathbf{c} - \mathbf{a}\utilde{\mathbf{d}})i}{\mathbf{c}^2 + \utilde{\mathbf{d}}^2}. \quad (7)$$

After that, we combine Eq. 6 and Eq. 7 to calculate the angle difference $\Delta\theta_{zw}$ between $\mathbf{z}$ and $\mathbf{w}$. By combing Eq. 6 and Eq. 7, we can obtain the following equation:

$$\frac{(\mathbf{ac} + \utilde{\mathbf{b}}\utilde{\mathbf{d}}) + (\utilde{\mathbf{b}}\mathbf{c} - \mathbf{a}\utilde{\mathbf{d}})i}{\mathbf{c}^2 + \utilde{\mathbf{d}}^2} = \gamma e^{i\Delta\theta_{zw}}. \quad (8)$$

Then, we apply $\log(\cdot)$ function to both sides, as follows:

$$\log(\frac{(\mathbf{ac} + \utilde{\mathbf{b}}\utilde{\mathbf{d}}) + (\utilde{\mathbf{b}}\mathbf{c} - \mathbf{a}\utilde{\mathbf{d}})i}{\mathbf{c}^2 + \utilde{\mathbf{d}}^2}) = \log(\gamma) + i\Delta\theta_{zw}. \quad (9)$$

Next, we move $\log(\gamma)$ to the left side and replace γ to $\frac{\sqrt{\mathbf{a}^2+\utilde{\mathbf{b}}^2}}{\sqrt{\mathbf{c}^2+\utilde{\mathbf{d}}^2}}$, as follows:

$$\log\left[\frac{(\mathbf{ac} + \utilde{\mathbf{b}}\utilde{\mathbf{d}}) + (\utilde{\mathbf{b}}\mathbf{c} - \mathbf{a}\utilde{\mathbf{d}})i}{\mathbf{c}^2 + \utilde{\mathbf{d}}^2} \times \frac{\sqrt{\mathbf{c}^2 + \utilde{\mathbf{d}}^2}}{\sqrt{\mathbf{a}^2 + \utilde{\mathbf{b}}^2}}\right] = i\Delta\theta_{zw}. \quad (10)$$

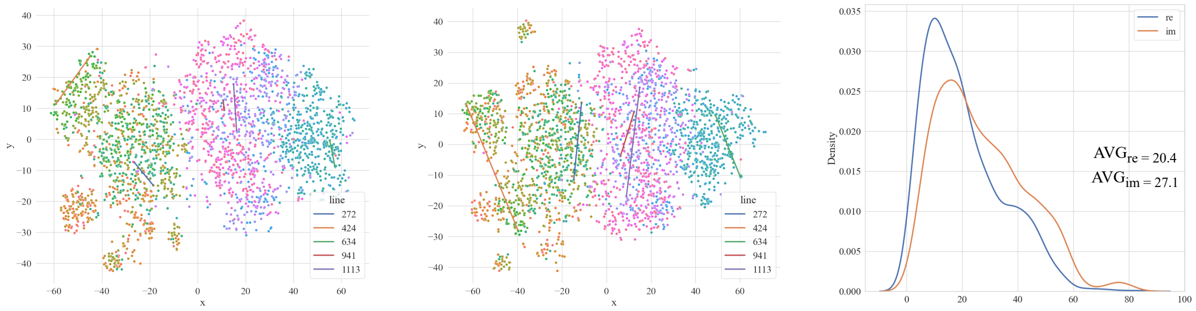
Finally, we simplify it and follow Sun et al. (2019) to use the real and imaginary text embeddings for the calculation to obtain Eq. 2

D Discussion

Discussion of Training Time. To evaluate the efficiency of AoE, we compare its training time with SBERT and SimCSE. We train the models on the STS-B dataset for one epoch using a single GPU (Nvidia GeForce RTX3090 Ti). For BERT$_{base}$, the training times are 14.35, 14.93, and 14.94 seconds for SBERT, SimCSE, and AoE, respectively. For LLaMA$_{7B}$, the training times are as follows: 1027.02 seconds for SBERT, 1027.67 seconds for SimCSE, and 1027.18 seconds for AoE. We find that AoE's training time is similar to SBERT and SimCSE, suggesting that AoE can achieve better performance with comparable efficiency.

Analysis of Real and Imaginary Text Embeddings in Saturation Zone. Figure 10 displays a 2D plot using t-SNE, showing the full STS-B test set's real and imaginary text embeddings. The imaginary text embeddings are more vertically scattered than the real text embeddings shown in Figure 10a and Figure 10b. The figures also include lines representing five sample text pairs, where the lines of the imaginary text embeddings are longer than those of the real text embeddings.

Figure 10c depicts the distance distribution of text pairs. It is noticeable that the distribution



(a) *Re* Text Embedding

(b) *Im* Text Embedding

(c) Distance Distribution

Figure 10: The t-SNE visualization of real (*Re*) and imaginary (*Im*) text embeddings and the kernel density estimate plot of the real and imaginary distance between text pairs in STS-B test set. AVG$_{re}$ and AVG$_{im}$ indicate the average distance between text pairs of the real and imaginary text embeddings.

of imaginary text embeddings is shifted towards higher distances compared to the real text embeddings. Moreover, the average distance of the imaginary text embeddings is also larger than that of the real text embeddings.

This evidence suggests that the imaginary text embeddings possess stronger capabilities in distinguishing semantic differences, thereby better discerning subtle semantic differences.

E Pretrained Models of AoE

We open source multiple AoE embeddings for various scenarios, as listed in Table 8. The universal AoE embeddings (UAE) can be used for information retrieval, retrieval-augmented generation (RAG), semantic textual similarity, code similarity, clustering, classification, and many other applications. The AoE NLI embeddings can be used for semantic textual similarity.