Message Passing on Semantic-Anchor-Graphs for Fine-grained Emotion Representation Learning and Classification

Pinyi Zhang[1,*], **Jingyang Chen**[2,*], **Junchen Shen**[1,*], **Zijie Zhai**[1,*], **Ping Li**[3,†], **Jie Zhang**[2], **Kai Zhang**[1,†]

[1]School of Computer Science and Technology, East China Normal University

[2]Institute of Science and Technology for Brain-inspired intelligence, Fudan University

[3]School of Computer Science and Software Engineering, Southwest Petroleum University

51265901020@stu.ecnu.edu.cn, 21110850022@m.fudan.edu.cn, 51215901048@stu.ecnu.edu.cn

51265901041@stu.ecnu.edu.cn, dping.li@gmail.com, jzhang080@gmail.com, kzhang980@gmail.com

Abstract

Emotion classification has wide applications in education, robotics, virtual reality, etc. However, identifying subtle differences between fine-grained emotion categories remains challenging. Current methods typically aggregate numerous token embeddings of a sentence into a single vector, which, while being an efficient compressor, may not fully capture their complex semantic and temporal distributions. To solve this problem, we propose SEmantic ANchor Graph Neural Networks (SEAN-GNN) for fine-grained emotion classification. It learns a group of representative, multi-faceted semantic anchors in the token embedding space: using these anchors as global reference, any sentence can be projected onto them to form a "semantic-anchor graph", with node attributes and edge weights quantifying semantic and temporal information, respectively. The graph structure is well aligned across sentences and, importantly, allows for generating comprehensive emotion representations regarding K different anchors. Message passing on the anchor graph can further integrate the semantic and temporal information and refine the learned features. Empirically, SEAN-GNN produces meaningful semantic anchors and discriminative graph patterns, with promising classification results on 6 popular benchmark datasets against state-of-the-arts.

1 Introduction

Emotion classification is an important task with applications in many fields like education, virtual reality, and robotics. However, fine-grained emotion classification (FEC) remains a challenging problem that is far from being well-solved. Unlike coarse-grained emotion classification (CEC), which may classify emotions into only a few basic categories (Ekman et al., 1999), FEC requires more detailed distinctions. For example, the two largest fine-grained emotion classification datasets contain 32 (Rashkin et al., 2019) and 27 (Demszky et al., 2020) categories, respectively.

The difficulty of fine-grained emotion classification mainly arises from learning faithful emotion representations, in particular in terms of capturing both the semantic and temporal distribution of emotion-related vocabulary in the sentence:

Semantically, human emotions are expressed by highly diverse word vocabulary (emotion-related adjectives, nouns, verbs and adverbs describing the intensity of the situation). Capturing the distribution of this rich vocabulary, and the subtle difference between similar emotions (e.g., afraid and terrified) is still an important challenge for fine-grained emotion classification.

Temporally, the meaning of a sentence is related to the meanings of its parts and the way they are combined (Pagin, 2016); in particular, subtle differences of emotion categories are in many cases presented by the relationship among the words (Waugh, 1977)[†]. Therefore capturing the temporal (or positional) word relations is crucial for emotion classification. *Unfortunately, since different sentences have different word compositions, directly quantifying and comparing word relationships across sentences is practically challenging*.

Numerous methods have been proposed for fine-grained emotion classification, see a review in Section 2. Despite the technical diversity, these methods typically use pre-trained language models (PLMs) or those enhanced with contrastive learning (Suresh and Ong, 2021) or LSTM (Zanwar et al., 2022) to obtain the token embeddings, and then aggregate them into a single vector for

[*]These authors contribute equally.

[†]Corresponding Author.

[†]Consider two sentences: "I feel extremely sad when I see animals abandoned and left to suffer." and "I feel sad when I see extremely pitiful animals abandoned and left to suffer." In the former, *extremely* describes *sad* and indicates a deeper emotion like devastated, while in the latter, *extremely* modifies *pitiful*, hence the emotion conveyed remains sadness.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2771–2783
November 12-16, 2024

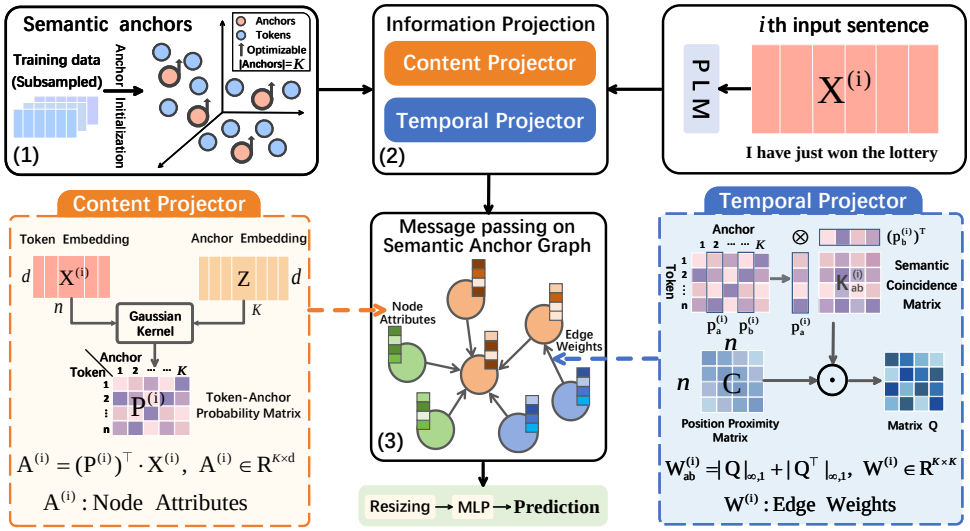

Figure 1: The structure of SEAN-GNN. (1) The K semantic anchors are learned end-to-end to cover emotion relevant vocabulary. (2) For an input sentence, the content-projector and the temporal projector are used to instill its semantic distribution and token relationship into an anchor graph. (3) A message passing GNN is used to integrate the semantic and temporal information and refine the anchor representations for final classification.

sentence-level representation. Well-known aggregating schemes include average pooling (Su et al., 2021), sum-pooling (Alvarez-Gonzalez et al., 2021), and [CLS] token (Sosea and Caragea, 2021; Suresh and Ong, 2021; Chen et al., 2023).

Although PLMs provide informative token embeddings for a sentence, aggregating them into a single vector may lead to significant information compression. From data distribution point of view, average pooling of the tokens is like approximating their distribution with the first-order statistics, and higher-order information (e.g. relationship among the tokens) may not be fully quantified[†]. However, the semantic distribution and the temporal relationship among the tokens are important information for accurate emotion classification.

In this paper, we explore new ways for computing sentence-level representations to capture complex semantic distributions and temporal relationship of the tokens. Unlike current methods that compress all the tokens of a sentence into one vector, we use a set of "semantic anchors" to extract sentence information in a more delicate manner. Our method is called semantic anchor graph neural network (SEAN-GNN), as in Figure 1.

The SEAN-network has three building blocks. (1) Learning semantic anchors, a set of vectors shared globally in the token embedding space covering emotion-related vocabulary. (2) Projecting a sentence onto the anchors by **content projector** (which projects token embeddings to anchors by their semantic similarity) and **temporal projector** (which projects the positional token relationship onto pairs of anchors). Then a sentence of arbitrary length can be expressed as a constant-sized anchor-graph, where node attributes and edge weights in turn quantify semantic and temporal information. (3)Using GNN to integrate semantic and temporal information to refine graph representations.

The semantic anchors provide a flexible and fine-grained basis for learning emotion representations. The anchors are learned end-to-end to cover emotion related vocabulary adaptively. Besides, anchors are shared globally, and so complex token relations from sentences of different word compositions, which are otherwise hard to compare, can now be easily quantified using the anchors as a common ground. This is beneficial since subtle positional relations of words can be important emotion features. Most importantly, rather than compressing all the tokens into a single vector, the semantic anchor graph allows one sentence to be encoded by multiple vectors each associated with one semantic anchor, being a highly enriched representation for fine-grained emotion classification.

Main contributions of the paper are listed below:

- We proposed SEAN-GNN to extract emotion-related features in a more delicate manner for fine-grained emotion classification.
- We show that SEAN-GNN learns meaningful

[†]The [CLS] embedding can be deemed as a weighted average of the token embeddings, so with similar observation.

semantic anchors and discriminative graph patterns for different emotion categories.

• We show that SEAN-GNN has promising results against state-of-the-arts across various base PLMs and 6 popular benchmark datasets.

2 Related Work

Numerous methods have been proposed for fine-grained emotion classification. Typically, pre-trained language models (PLM) are used to get token embeddings; these embedding are then further refined/updated before aggregated into a single vector for emotion classification. Various strategies were designed to refine either of these 3 steps.

For PLMs, Sosea and Caragea (2021) presented emotion masked language modeling, which only masked off those emotion-related tokens in the pre-training stage; Yin and Shang (2022) incorporated a whitening method and nearest neighbor retrieval to PLMs to improve retrieval efficiency so as to better differentiate semantically similar sentences.

For token embedding refinement, Suresh and Ong (2021) modified the supervised contrastive loss (Khosla et al., 2020) and propose a label-aware contrastive loss to improve token embedding; Chen et al. (2023) proposed HypEmo, which learned label embedding in hyperbolic space and integrated it with RoBERTa fine-tuned in Euclidean space.

For token aggregation, a common method involves pooling all token embeddings into a single vector through averaging, summation, or taking the maximum/minimum. For instance, Zanwar et al. (2022) employed Bi-LSTM to process token embeddings derived from a pre-trained language model, and concatenated the hidden representations from Bi-LSTM's final layer to form the context representation. Alvarez-Gonzalez et al. (2021) suggested that a pooling function such as attention, mean or max can be used to aggregate token embeddings to a vector; The [CLS] token embedding is also commonly used as the sentence-level feature (Devlin et al., 2018). However, Su et al. (2021) showed that averaging the token embeddings is better than only utilizing [CLS] token. Both are sub-optimal as demonstrated by Choi et al. (2021).

Despite the technical diversity, most of these methods mix up the token embeddings of a sentence into a single vector. Such aggregation is a convenient way for sentence-level representation, but it may not be sufficiently effective in capturing the semantic and temporal distribution, which can be crucial to accurate emotion classification.

In the NLP literature, concept of anchors have been explored in various tasks, but with motivations and implementations very different from our approach. For example, Arora et al. (2012) selected words that are uniquely associated with a topic as anchors to accelerate topic modeling analysis. Liu et al. (2020) adopted the average contextual representations of each word as the anchors to enhance contextualized representations. Wang et al. (2023) used the class labels/words as anchors and used anchor re-weighting to improve in-context learning performance. In these works, anchors are linked to predefined words, while our anchors are learned adaptively through data.

GNN models have also been applied in NLP tasks, like encoding word relations (Yao et al., 2019), recognizing named entities (Luo and Zhao, 2020), modeling syntactic structures (Luo and Zhao, 2020), etc. A main difference is that our GNN is built on *semantic anchors* rather than raw tokens. Using anchors as GNN nodes allows generating emotion representations that are not only rich and multi-faceted, but also well-aligned across different sentences without token padding or cutting (an undesired perturbation of their embeddings).

GNN models themselves could also benefit from the use of anchors. These methods use anchors to improve the computational efficiency of GNNs or graph-based clustering/semi-supervised learning (Liu et al., 2010; Nie et al., 2022; You et al., 2019, etc), to better encode relative positional relation between the nodes (You et al., 2019), or to improve graph embedding in case of noisy/inaccurate edges (Tu et al., 2022). These anchor based GNN models are different from ours in both their motivations and methodology. They mainly consider graphs like similarity graph (clustering), protein networks and communication networks (link prediction and community detection), without temporal (sequential) relation between the nodes; in comparison, a challenge in our context is how to properly project the temporal relationship between pairs of tokens onto their corresponding anchors. Furthermore, our anchors are learned end-to-end, instead of being directly selected from existing nodes or computed through an off-line procedure.

3 Methodology

The SEAN-GNN model has three main modules, as discussed in the following three subsections.

3.1 Semantic Anchors

Given m sentences each shaped to the same length of n tokens, as $\mathbf{X}^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, ..., \mathbf{x}_n^{(i)}\}$. Here $\mathbf{x}_j^{(i)} \in \mathbb{R}^{d\times 1}$ is the embedding of the jth token of the ith sentence. In order to account for the diversity of emotion-related vocabulary in the training data, we propose to learn a global semantic anchor set, $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_K\}$ to facilitate the representation learning for emotion classification. Each $\mathbf{z}_k \in \mathbb{R}^{d\times 1}$ is a vector in the word embedding space. Preferably, the learned anchors should be diverse enough to cover different emotional aspects, while in the mean time discriminative enough to generate good features for accurate classification.

To promote diversity of semantic anchors, we collect token embeddings from all (or a random subset of) the input sentences obtained through pretrained language model, and then initialize the anchors as their K-means clustering centers. The K-means algorithm is known to distribute clustering centers to minimize the reconstruction error of the input samples. We further optimize the semantic anchors in the end-to-end architecture in Figure 1. By doing this, the semantic anchors will be iteratively optimized and updated to facilitate extraction of discriminative semantic and temporal features for emotion classification.

3.2 Information Projection through Semantic Anchor Graph

Using the K anchors $\{\mathbf{z}_k's\}$ as global reference, we can project the information of sentence $\mathbf{X}^{(i)}$ onto it and obtain a graph representation as $\mathcal{G}^{(i)} = (\mathbf{A}^{(i)}, \mathbf{W}^{(i)})$. We call $\mathcal{G}^{(i)}$ the *semantic-anchor graph* (SEAN-graph) for sentence $\mathbf{X}^{(i)}$, which has exactly K nodes corresponding to the K anchors. The node attribute matrix $\mathbf{A}^{(i)} \in \mathbb{R}^{K\times d}$ and adjacency matrix $\mathbf{W}^{(i)} \in \mathbb{R}^{K\times K}$ respectively encodes the semantic (first-order) and the temporal (second-order) distribution of the input sentence. Since anchors are shared across sentences with wide coverage and discriminative power, the semantic anchor graph $\mathcal{G}^{(i)}$ serves as an informative and well-aligned emotion representation.

To project the input sentence $\mathbf{X}^{(i)}$ onto the K anchors to extract its semantic/temporal information, we have devised the following two projectors:

- Content projector. The semantics/embeddings of the words of a sentence are projected as node attributes ($\mathbf{A}^{(i)}$) of the SEAN-graph.
- Temporal projector. The temporal relations between the words of a sentence are projected as edge weights ($\mathbf{W}^{(i)}$) of the SEAN-graph.

Content projector. Suppose we are given an input sentence with token embedding matrix $\mathbf{X}^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, ..., \mathbf{x}_n^{(i)}\}$. The goal of the content projector is to project each token to the K anchors $\mathbf{z}_k$'s in a probabilistic manner. We use a probability matrix $\mathbf{P}^{(i)} \in \mathbb{R}^{n\times K}$ whose jkth entry denote the probability that the jth token in the ith sentence belongs to the kth anchor, such that

$$\mathbf{P}_{jk}^{(i)} = \frac{\exp\left(-\frac{\|\mathbf{x}_j^{(i)} - \mathbf{z}_k\|^2}{2\sigma^2}\right)}{\sum_{k=1}^{K}\exp\left(-\frac{\|\mathbf{x}_j^{(i)} - \mathbf{z}_k\|^2}{2\sigma^2}\right)} \quad (1)$$

with σ the bandwidth of the Gaussian. In other words, each row of $\mathbf{P}^{(i)}$ specifies the probability of one token belonging to the K anchors. It can also be deemed as the cross-attention matrix between tokens and anchors. After quantifying the probabilistic association between n tokens and the K anchors, we can project the token embeddings onto the anchors as,

$$\mathbf{A}^{(i)} = (\mathbf{P}^{(i)})^\top \cdot \mathbf{X}^{(i)}. \quad (2)$$

The matrix $\mathbf{A}^{(i)} \in \mathbb{R}^{K\times d'}$ can be used as the attribute matrix of the semantic anchor-graph $\mathcal{G}^{(i)}$. Intuitively, the kth row in $\mathbf{A}^{(i)}$ summarizes the content of the sentence that are most relevant to the kth semantic anchor. If the tokens are all irrelevant to that anchor, the kth row of $\mathbf{A}^{(i)}$ approaches 0.

Temporal Projector. The goal of the temporal projector is to project the temporal/positional relation between pairs of tokens in a sentence onto pairs of anchors. This allows token relationship in each sentence to be expressed globally as the relationship among the K semantic anchors.

Suppose we have projected a sentence $\mathbf{X}^{(i)}$ onto K anchors, with the token-anchor probability matrix $\mathbf{P}^{(i)}$ (1). We normalize it such that kth column in $\mathbf{P}^{(i)}$ becomes a probability simplex describing the probabilities that a word similar to the kth anchor appears in the n locations of the sentence $\mathbf{X}^{(i)}$. It can be deemed as the positional distribution of the kth anchor in the sentence. We will use these columns to evaluate the relations between any pair of anchors for input sentence $\mathbf{X}^{(i)}$, as follows.

Let $\mathbf{p}_a^{(i)}$ and $\mathbf{p}_b^{(i)}$ denote two columns of $\mathbf{P}^{(i)}$, i.e., $\mathbf{p}_a^{(i)} = \mathbf{P}_{[:,a]}^{(i)}$, $\mathbf{p}_b^{(i)} = \mathbf{P}_{[:,b]}^{(i)}$, as illustrated in

Figure 2. For each of the n entries/locations in $\mathbf{p}_a^{(i)}$, say, the sth entry with (large) probability $\mathbf{p}_a^{(i)}(s)$, we will examine the entries inside the location window $[s-l, s+l]$ in the probability vector $\mathbf{p}_b^{(i)}$. If there exists a large probability in this window, that means two words whose meanings are similar to the two anchors respectively appear in close vicinity to each other within the input sentence $\mathbf{X}^{(i)}$. This should contribute positively to the temporal relation between the two anchors. We will examine all the entries in $\mathbf{p}_a$ and accumulate the scores. Mathematically, the temporal relation between the ath anchor and the bth anchor due to the input sentence $\mathbf{X}^{(i)}$ can then be computed as follows,

$$\mathbf{W}_{ab}^{(i)} = \sum_{s=1}^{n} \mathbf{p}_a^{(i)}(s) \sum_{t=1}^{n} \mathbf{p}_b^{(i)}(t) \cdot \exp\left(-|s-t|\right)$$

It can be deemed as the correlation between two probability simplex vectors (positional distribution of two anchors) but with relaxed positional alignment. Note that if we scan through entries in $\mathbf{p}_b^{(i)}$ and find neighbors in $\mathbf{p}_a^{(i)}$, the resultant, accumulated score will be the same (see Appendix A.1).

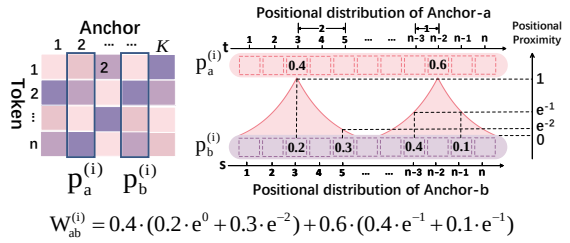

Figure 2: The temporal relation between two anchors, a and b, for input sentence $\mathbf{X}^{(i)}$ based on their respective positional distributions in this sentence.

The $\mathbf{W}_{ab}^{(i)}$ can be computed in matrix form as follows. Define the $n \times n$ probabilistic coincidence matrix using outer-product $\mathbf{K}_{ab}^{(i)} = \mathbf{p}_a^{(i)}(\mathbf{p}_b^{(i)})^\top$, and $n \times n$ positional proximity matrix $\mathbf{C}$ such that $\mathbf{C}_{st} = \exp(-|s-t|)$. Then $\mathbf{W}_{ab}^{(i)}$ can be computed as the sum of entries of the hadamard product

$$\mathbf{W}_{ab}^{(i)} = \left|\mathbf{K}_{ab}^{(i)} \odot \mathbf{C}\right|_1 \tag{3}$$

Empirically, revising the ℓ_1-norm in (3) to the mixed-norm $\ell_{\infty,1}$ (summation of the maximum entry of each row) gives more robust result. This means that for a token in one of the two positional distributions, $\mathbf{p}_a^{(i)}$ and $\mathbf{p}_b^{(i)}$, we emphasize only the most significant word pairs across the two distributions. This, however, breaks the symmetry so we have to compute symmetric version

$$\mathbf{W}_{ab}^{(i)} = \left|\mathbf{K}_{ab}^{(i)} \odot \mathbf{C}\right|_{\infty,1} + \left|(\mathbf{K}_{ab}^{(i)})^\top \odot \mathbf{C}\right|_{\infty,1} \tag{4}$$

3.3 Message Passing on the Anchor-Graph

Having encoded the semantic/temporal information of sentence $\mathbf{X}^{(i)}$ as an undirected anchor graph $\mathcal{G}^{(i)}$, with node attribute matrix $\mathbf{A}^{(i)}$ and adjacency matrix $\mathbf{W}^{(i)}$, we employ GNNs (Kipf and Welling, 2016; Velickovic et al., 2017; Hamilton et al., 2017, etc.) to perform message passing among the anchor nodes. The procedures using GCN (Kipf and Welling, 2016) is as follows.

$$\begin{aligned} \mathbf{H}^{[l]} &= \mathbf{A}^{(i)} \\ \tilde{\mathbf{W}} &= \mathbf{W}^{(i)}, \;\; \tilde{\mathbf{D}}_{ii} = \sum\nolimits_j \tilde{\mathbf{W}}_{ij} \\ \mathbf{H}^{[l+1]} &= \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{W}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{[l]}\mathbf{\Theta}^{[l]}\right) \end{aligned} \tag{5}$$

Here, $\mathbf{H}^{[l]}$ is node feature matrix at layer l, and the 0th layer is initialized by $\mathbf{A}^{(i)}$; $\tilde{\mathbf{D}}$ is normalized degree matrix, $\tilde{\mathbf{W}}$ is chosen as the adjacency matrix between anchors, σ is the ReLU (Krizhevsky et al., 2012), and $\mathbf{\Theta}^{[l]}$is the transform at layer l.

The message passing on semantic anchor graph will aggregate the features of those anchor-nodes having close temporal relations with each other according to the input sentence. In other words, the temporal and semantic information of the input sentence are integrated through GNN to enhance the anchor features, and the resultant attribute matrix $\mathbf{H}^{[l]}$ will uniquely determine representation of the input sentence. We concatenate $\mathbf{H}^{[0]}$ and final-layer $\mathbf{H}^{[l]}$ as the sentence-level representation, and flatten it to a long vector with a 3-layer FFN and cross entropy loss for classification. Note that our method allows any GNN model for message passing, lending itself great flexibility in applications.

4 Experiments

4.1 Datasets

We evaluate our model on altogether 6 benchmark datasets widely used for emotion classification. Among them, the first two are fine-grained classification (the two largest and most challenging datasets we could find in the literature), and the other 4 datasets are course-grained classification. The way we pre-process each dataset follows previous works (Chen et al., 2023; Suresh and Ong, 2021; Yin and Shang, 2022). Brief data statistics are listed below (see more details in Appendix B).

(1) **Empathetic Dialogue** (Rashkin et al., 2019) consists of dialogues between a speaker and a listener with 32 single emotion label.
(2) **GoEmotion** (Demszky et al., 2020) are Reddit comments from 27 emotions and neutral.
(3) **CancerEmo** (Sosea and Caragea, 2020) composes of 8500 sentences sampled from an online cancer survivors network with 8 emotion labels.
(4) **ISEAR** (Scherer and Wallbott, 1994) contains sentences of personal reports on emotional events labelled with one of 7 emotions.
(5) **GoEmotion-EK** (Ekman et al., 1999) annotates data originally constructed by (Demszky et al., 2020) into Ekman's 6 basic emotions.
(6) **EmoInt** (Mohammad and Bravo-Marquez, 2017) comprises tweets of 4 emotion classes.

4.2 Experimental Settings and Baselines

Metrics. For fine-grained emotion classification, we adopt Accuracy and Weighted F1 by following the setting in (Suresh and Ong, 2021) and (Chen et al., 2023). For coarse-grained emotion classification, we use Macro F1 following the common practice in (Yin and Shang, 2022; Singh et al., 2023).

Baselines. We incorporated 12 baseline methods. Baseline methods (1-6) are three PLMs (BERT, RoBERTa, and ELECTRA) with two sizes (base, large), all using the [CLS] token embedding as the sentence-level feature. The remaining 6 baselines are recent state-of-the-art methods, including: (7) **LCL** (Suresh and Ong, 2021) using label-aware contrastive loss; (8) **Hypemo** (Chen et al., 2023), using label-aware weighting and hyperbolic distance metric; (9-10) **PLM-BiLSTM** and **PLM-DNN** (Alvarez-Gonzalez et al., 2021) using Bi-LSTM and DNN to update token embeddings from PLMs with summation pooling; (11) **PsyLing** (Zanwar et al., 2022) use Bi-LSTM trained on psycholinguistic features to improve the generalizability for emotion classification of token embeddings from PLMs; (12) **KNNEC** (Yin and Shang, 2022) using whitening method and nearest neighbor retrieval for emotion classification.

Method (7-12) and ours need a base PLM to compute token embeddings. For fairness of comparison, we used $\text{RoBERTa}_{\text{base}}$ for all, which was also the majority of their official choices. For those officially reported results using $\text{BERT}_{\text{base}}$, our comparisons with them are in Appendix C.1.

Our algorithm used batch size 64 and AdamW optimizer, with a learning rate $2e^{-5}$ and a weight decay 0.01. Graph convolutional network (Kipf and Welling, 2016) is used for message passing. The number of semantic anchor K was chosen from {50, 100, 150, 200} using validation set. The parameter settings of other methods follow their original papers, see details in Appendix C.2.

4.3 Classification Results

Results are reported in Table 1. Each evaluation is based on 5 repeats of different seeds, with the average score and standard deviation.

Our results surpassed over PLMs (base and large versions), with weighted-F1 being 4.0% and 2.2% higher than the best among them ($\text{RoBERTa}_{\text{large}}$) in two fine-grained classifications. In the 4 coarse-grained tasks, our model surpassed $\text{RoBERTa}_{\text{large}}$ in Macro-F1 by 1.1% - 2.9%. Note that our model was only based on the base version of RoBERTa. Therefore these performance gains are clearly attributed to the use the semantic anchor graph in aggregating token embeddings.

Our model also outperforms other advanced algorithms with an improvement of 1.2% and 1.1% in weighted F1 , 1.6% and 2.2% in accuracy against the best competitor on 2 fine-grained tasks; on 4 coarse-grains datasets, an improvement around 1.1% - 2.2% in Macro F1 was observed. Overall, our method has shown promising results across all metrics and datasets.

4.4 Impact of Base PLMs and Anchor-set Size

Table 2 reports the results of our method *using anchor-based sentence features* when the raw token embeddings are obtained from different base PLMs ($\text{BERT}_{\text{base}}$, $\text{RoBERTa}_{\text{base}}$, $\text{ELECTRA}_{\text{base}}$). It also reports the results of these PLMs *using [CLS] embedding as sentence features*. As can be seen, SEAN-GNN can enhance performance irrespective of the PLM employed, with an improvement around 3.3% - 9.4%. This shows that our approach is PLM-agnostic and can be versatilely integrated with any PLMs to improve performance.

We also investigate how the number of semantic anchors, K, affects the performance. Using the two fine-grained datasets, we plot the Weighted F1 score of our method when K is chosen from 1 to 500. Here, K=1 can be deemed as the standard pooling. As shown in Figure 3, the performance exhibits a significant improvement when K increases from 1 to 100, validating the effectiveness of introducing semantic anchors to emotion classification. When K increases to 200, the performance remains steady, meaning that the gains due to larger num-

	Empathetic Dialogue 32		GoEmotions 27		CE 8	IS 7	EK 6	EM 4
	Acc	Weighted F1	Acc	Weighted F1	Macro F1			
$\text{BERT}_{\text{base}}$	50.4 ± 0.3	51.8 ± 0.21	60.9 ± 0.4	62.9 ± 0.5	70.1 ± 1.4	69.2 ± 0.8	71.1 ± 1.1	84.8 ± 0.6
$\text{RoBERTa}_{\text{base}}$	54.5 ± 0.7	56.0 ± 0.4	62.6 ± 0.6	64.0 ± 0.2	73.6 ± 1.3	69.4 ± 0.9	71.9 ± 0.7	85.4 ± 0.6
$\text{ELECTRA}_{\text{base}}$	47.7 ± 1.2	49.6 ± 1.0	59.5 ± 0.4	61.6 ± 0.6	72.1 ± 0.5	69.9 ± 1.2	71.4 ± 1.3	85.2 ± 0.9
$\text{BERT}_{\text{large}}$	53.8 ± 0.1	54.3 ± 0.1	64.5 ± 0.3	65.2 ± 0.4	72.3 ± 0.7	70.2 ± 1.4	71.6 ± 0.9	85.6 ± 0.5
$\text{RoBERTa}_{\text{large}}$	57.4 ± 0.5	58.2 ± 0.3	64.6 ± 0.3	65.2 ± 0.2	74.7 ± 1.0	73.1 ± 0.5	73.0 ± 0.8	86.0 ± 0.7
$\text{ELECTRA}_{\text{large}}$	56.7 ± 0.6	57.6 ± 0.6	63.5 ± 0.2	64.1 ± 0.3	73.5 ± 0.9	72.5 ± 1.4	72.0 ± 0.7	85.3 ± 0.7
PLM-BiLSTM†	55.3 ± 1.1	56.9 ± 0.9	63.4 ± 1.4	64.6 ± 0.8	73.8 ± 0.7	69.9 ± 1.2	72.3 ± 0.6	85.6 ± 0.6
PLM-DNN†	55.1 ± 0.7	57.2 ± 1.3	63.0 ± 0.5	64.3 ± 1.4	74.4 ± 0.9	70.3 ± 1.1	72.6 ± 0.8	85.4 ± 0.5
PsyLing	56.5 ± 1.2	57.0 ± 1.1	63.0 ± 0.6	64.6 ± 1.3	74.7 ± 0.7	71.7 ± 1.4	73.0 ± 0.9	85.7 ± 0.3
KNNEC	58.0 ± 0.9	58.5 ± 0.8	64.0 ± 1.2	64.5 ± 1.0	74.4 ± 0.6	70.7 ± 1.1	73.5 ± 1.3	86.0 ± 0.7
LCL†	59.5 ± 0.6	59.2 ± 0.5	64.5 ± 0.3	65.1 ± 0.3	75.0 ± 0.8	72.1 ± 1.0	72.8 ± 1.2	86.3 ± 0.3
HypEmo	59.6 ± 0.3	61.0 ± 0.3	65.4 ± 0.2	66.3 ± 0.2	75.4 ± 0.6	73.0 ± 1.4	73.2 ± 0.8	86.0 ± 0.6
Ours	**61.2 ± 0.3**	**62.2 ± 0.2**	**67.6 ± 0.4**	**67.4 ± 0.5**	**77.6 ± 0.3**	**74.2 ± 0.6**	**74.7 ± 0.5**	**87.6 ± 0.4**
Δ	+ 1.6%	+ 1.2%	+ 2.2%	+ 1.1%	+ 2.2%	+ 1.1%	+ 1.2%	+ 1.3%

Table 1: Classification results (in %) for all methods, with weighted F1 and accuracy for fine-grained task (Suresh and Ong, 2021; Chen et al., 2023), and Macro F1 for coarse-grained task (Yin and Shang, 2022; Singh et al., 2023). The best/second-best results highlighted in **bold**/ underline. "†" indicates we present results using $\text{RoBERTa}_{\text{base}}$ as backbone for fairness. CE, IS, EK, EM stands for CancerEMO, ISEAR, GoEmotion-EK, EmoInt; numerals are the number of classes. Δ represents the improvement of our model over the second-best.

Dataset	PLM	w/o	with	Δ
ED	$\text{BERT}_{\text{base}}$	51.8	**58.8**	+ 7.0%
ED	$\text{RoBERTa}_{\text{base}}$	56.0	**62.2**	+ 6.2%
ED	$\text{ELECTRA}_{\text{base}}$	49.6	**59.0**	+ 9.4%
GE	$\text{BERT}_{\text{base}}$	62.9	**66.2**	+ 3.3%
GE	$\text{RoBERTa}_{\text{base}}$	64.0	**67.4**	+ 3.4%
GE	$\text{ELECTRA}_{\text{base}}$	61.6	**64.9**	+ 3.3%

Table 2: Weighted F1 score of different PLMs using [CLS] token as sentence embedding, and that using SEAN-GNN for sentence embedding. ED for Empathetic Dialogue and GE for GoEmotion.

ber of anchors diminish. When K is larger than 300, the performance drops slightly by 0.5% - 1%. This is because too many additional anchors (beyond what is necessary) may lead to overfitting or introduce unnecessary noise. In practice, we use validation set to determine the number of anchors, which is typically around 100.

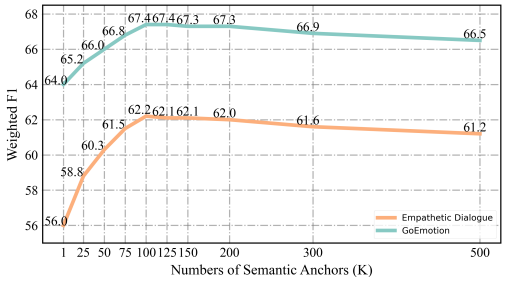

Figure 3: How the number of semantic anchors, K, affects the performance of SEAN-GNN.

4.5 Case Study

In this subsection, we examine whether SEAN-GNN can generate meaningful semantic anchors for emotion classification, as well as unique graph patterns for different emotion classes. Moreover, we report comparative results using 4 most difficult subsets of Empathetic Dialogues to further demonstrate the effectiveness of our method.

We choose three pairs of emotion classes with subtle difference: {Afraid vs Terrified}, {Angry vs Furious} and {Sad vs Devastated}. First, we pull out top-6 semantic anchors most relevant to each emotion, and annotate the anchor with two words with closest embeddings to it (see Appendix A.2). As shown in Figure 4, the learned anchors encompass verbs (run, shout, cry), nouns (murder, betrayal, despair) and adjectives (severe, unfair, upset). Their semantics look quite reasonable with each emotion class, like Terrified: {murder, crime, scream, frighten}, and Furious: {disrespect,insult}. Interestingly, for the intense emotion in each pair (e.g., furious, terrified), they are often associated with anchors of adverbs such as {so, really, very, quite}, which are absent in less-intense emotions (afraid, sad). Intense emotions may also employ anchors like {murder, yell} to describe the fierce state. These observations are consistent with our understanding of the emotions from a natural language perspective. A longer list is in Appendix A.3.

Figure 4 visualizes the averaged adjacency matrix (4) (edges with the top-10% highest weights)

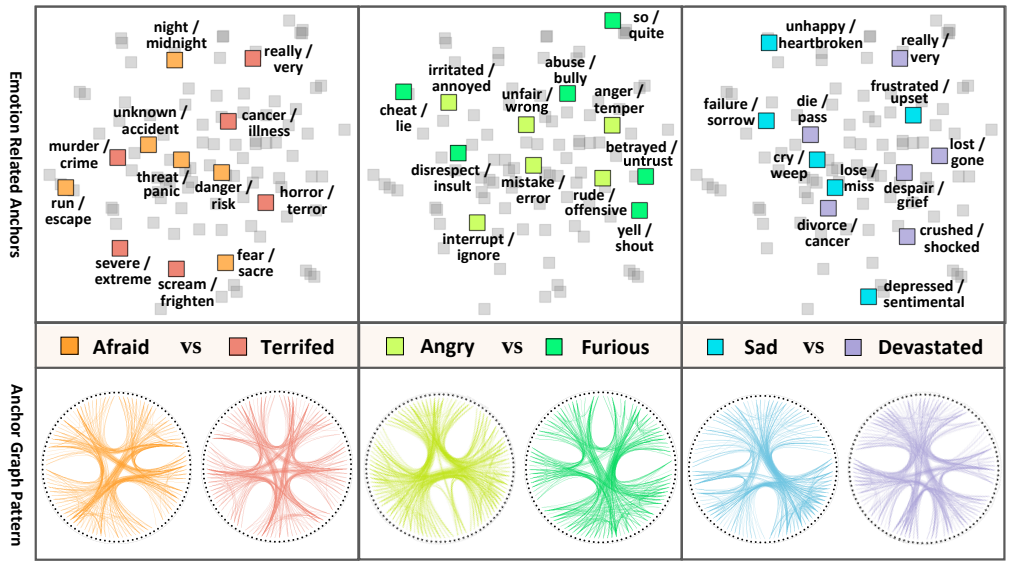

Figure 4: Visualization of semantic anchors (top row) and anchor-graph patterns (bottom row) learned by SEAN-GNN for 6 (3 pairs of easily confused) emotion classes. Top row: 6 most relevant anchors, each annotated by 2 closest words, for different emotions as visualized by tSNE (colored squares: class-relevant anchors; gray: less relevant). Bottom row: averaged anchor-graph patterns ($K \times K$ adjacency matrix in (4)) for each emotion class.

	subset_a	subset_b	subset_c	subset_d
$\text{RoBERTa}_{\text{base}}$	57.1	64.4	55.5	79.4
LCL	58.7	66.3	57.2	80.2
HypeEmo	63.6	69.5	60.0	81.1
Ours	**64.9**	**70.6**	**61.6**	**82.5**
Δ	1.3%	1.1%	1.6%	1.4%

Table 3: Weighted F1 (%) on 4 most confusable subsets of Empathetic Dialogue compared with previous effective methods and $\text{RoBERTa}_{\text{base}}$. Δ represents the improvement of our model over the second-best.

Dataset	Model	Weighted F1	Δ
ED	Complete	**62.2**	-
ED	w/o Te	60.2	- 2.0%
ED	w/o Te, GNN	58.4	- 3.8.%
ED	w/o Te, GNN, Se	56.0	- 6.2%
GE	Complete	**67.4**	-
GE	w/o Te	66.3	- 1.1%
GE	w/o Te, GNN	65.3	- 2.1%
GE	w/o Te, GNN, Se	64.0	- 3.4%

Table 4: Weighted F1 (%) on ED and GE datasets after sequentially removing the core component of our model. TP/SP: Temporal/Semantic Projector. Δ: the adverse impact due to removal of current component(s).

for sentences in each emotion category. The anchor-graph patterns show a clear difference even among emotion categories with only small difference.This shows the discriminative power of the anchor-graph based sentence representations.

Table 3 reports comparative results on four most confusable subsets of Empathetic Dialogue selected by Suresh and Ong (2021) (see Appendix D for details). Our method outperforms state-of-the-art methods by 1.1%-1.6% in weighted F1.

4.6 Ablation Study

SEAN-GNN has several core components: Content Projector, Temporal Projector, and GNN-module. We sequentially remove each component and report results for the two fine-grained datasets in Table 4 in Weighted F1. It is observable that the elimination of any one of the three components has a significant detrimental effect on the performance.

5 Conclusion

We proposed SEAN-GNN to extract the content distribution and toke relation for fine-grained emotion classification. It allows generating comprehensive and discriminative emotion representations, and has produced promising results across different benchmark datasets and base PLM embedding.

6 Acknowledgement

This work was supported in part by the National Key Research and Development Program of China (2022YFC3400501), National Natural Science Foundation of China (62276099), and East China Normal University Graduate Student Special Fund for International Conferences.

7 Limitations

We only evaluated the performance of the competing methods on datasets in the English, due to the lack of fine-grained emotion classification datasets in languages other than English,which potentially introduced language and cultural biases. Moreover, the risk of reinforcing existing data biases and the consideration of model fairness across different demographic groups were not addressed.

References

Nurudin Alvarez-Gonzalez, Andreas Kaltenbrunner, and Vicenç Gómez. 2021. Uncovering the limits of text-based emotion detection. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2560–2583.

Sanjeev Arora, Rong Ge, and Ankur Moitra. 2012. Learning topic models – going beyond svd. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 1–10.

Chih Yao Chen, Tun Min Hung, Yi-Li Hsu, and Lun-Wei Ku. 2023. Label-aware hyperbolic embeddings for fine-grained emotion classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10947–10958.

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*, pages 5482–5487. IEEE.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054.

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*.

Paul Ekman et al. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2023. Ambiguity-aware in-context learning with large language models. *arXiv preprint arXiv:2309.07900*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2020. Towards better context-aware lexical semantics: Adjusting contextualized representations through static anchors. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4066–4075.

Wei Liu, Junfeng He, and Shih-Fu Chang. 2010. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 679–686. Citeseer.

Zhiwei Liu, Kailai Yang, Qianqian Xie, Tianlin Zhang, and Sophia Ananiadou. 2024. Emollms: A series of emotional large language models and annotation tools for comprehensive affective analysis. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5487–5496.

Ying Luo and Hai Zhao. 2020. Bipartite flat-graph network for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6408–6418.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. WASSA-2017 shared task on emotion intensity. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, Copenhagen, Denmark.

Feiping Nie, Chaodie Liu, Rong Wang, Zhen Wang, and Xuelong Li. 2022. Fast fuzzy clustering based on anchor graph. *IEEE Transactions on Fuzzy Systems*, 30(7):2375–2387.

Peter Pagin. 2016. *Sentential semantics*, page 65–105. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Klaus R Scherer and Harald G Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310.

Gargi Singh, Dhanajit Brahma, Piyush Rai, and Ashutosh Modi. 2023. Text-based fine-grained emotion prediction. *IEEE Transactions on Affective Computing*.

Tiberiu Sosea and Cornelia Caragea. 2020. CancerEmo: A dataset for fine-grained emotion detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8892–8904, Online. Association for Computational Linguistics.

Tiberiu Sosea and Cornelia Caragea. 2021. emlm: a new pre-training objective for emotion related tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 286–293.

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

Varsha Suresh and Desmond Ong. 2021. Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4381–4394.

Enmei Tu, Zihao Wang, Jie Yang, and Nikola Kasabov. 2022. Deep semi-supervised learning via dynamic anchor graph embedding in latent space. *Neural Networks*, 146:350–360.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat*, 1050(20):10–48550.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855.

Linda R Waugh. 1977. *A semantic analysis of word order: Position of the Adjective in French*, volume 1. Brill Archive.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Wenbiao Yin and Lin Shang. 2022. Efficient nearest neighbor emotion classification with bert-whitening. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4738–4745.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31.

Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. In *International conference on machine learning*, pages 7134–7143. PMLR.

Sourabh Zanwar, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. Improving the generalizability of text-based emotion detection by leveraging transformers with psycholinguistic features. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, pages 1–13.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*.

A Appendix

A.1 Temporal Relation between two Anchors

We prove that if we scan the entries in $\mathbf{p}_a^{(i)}$ and find neighbors in $\mathbf{p}_b^{(i)}$, or if we do the opposite, the resultant score between the two anchors (a and b) will be the same, i.e., i.e.,

$$\begin{aligned} & \sum_{s=1}^{n} \mathbf{p}_a^{(i)}(s) \sum_{t=1}^{n} \mathbf{p}_b^{(i)}(t) \cdot \exp\left(-|s-t|\right) \\ = & \sum_{s=1}^{n} \mathbf{p}_b^{(i)}(s) \sum_{t=1}^{n} \mathbf{p}_a^{(i)}(t) \cdot \exp\left(-|t-s|\right) \end{aligned}$$

To prove this, we can simply swap the two summation indices, s and t; due to the exchangability of the two indices and that $\exp\left(-|s-t|\right) = \exp\left(-|t-s|\right)$, we can easily see the equivalence.

Note that the computation in (3) can also be written in quadratic terms $\mathbf{W}^{(i)} = (\mathbf{P}^{(i)})^{\top}\mathbf{C}\mathbf{P}^{(i)}$ considering that all the numbers are non-negative. This is computationally very efficient because all the pairwise anchor relations can be computed with two matrix multiplications. However, this may not be applicable to the computation of (4) because we

have to explicitly compute the hadamard matrix $\mathbf{K}_{ab}^{(i)} \odot \mathbf{C}$. This also makes our computation (of the information projector) very different from other methods in the literature such as the hierarchical pooling (Ying et al., 2018, etc.).

A.2 Identifying Emotion-Relevant Anchors

When learning the semantic anchors $\{\mathbf{z}'_k s\}$, they are not specifically tied to emotion classes but instead learned globally. After obtaining the anchors, however, we can associate each anchor to all the emotion classes so as to make post-hoc analysis.

Suppose we have m sentences, each with the feature $\mathbf{H}^{(i)} \in \mathbb{R}^{K \times d'}$ as learned by SEAN-GNN. Each sentence is also linked to a label vector $y^{(i)} \in \mathbb{R}^{1 \times L}$, with L the number of emotions. Then we can perform association analysis as follows. We flatten each $\mathbf{H}^{(i)}$ to a Kd'-dimensional vector, and put this vector all all the m sentences together as an $m \times Kd$ matrix; we also put the label vectors together and form a $m \times L$ matrix. We can then compute the correlation between these two matrices and obtain an $Kd' \times L$ association matrix. We compute the absolute value of this matrix and compress it to a $K \times L$ matrix by summing up those rows that belong to the same anchor. This matrix then tells the relevance between each anchor and each emotion class.

A.3 List of Anchors for Some Emotions

In Table 5, we report a longer list of the anchors that are associated with each emotion class, by choosing top 6 anchors for each class, (three closest words to each anchor for annotation), and 10 different emotion classes appearing in Empathetic Dialogue. In the following, we denote each anchor as (w_1, w_2, w_3), the three words with the closest embeddings to this anchor.

Emotion	Anchors
Afraid	(accident, crash, incidents), (night, midnight, dark), (danger, risk, hazard), (threat, panic, alarm), (run, escape, flee), (fear, scare, worry)
Terrified	(cancer, illness, disease), (severe, extreme, intense), (horror, terror, fear), (murder, crime, violence), (really, very, truly), (scream, frighten, shout)
Angry	(temper,rage,anger), (unfair,bullied,oppressed), (rude,offensive,impolite), (interrupt,ignore,disrupt), (mistake,error,fault), (irritated,annoyed,upset)
Furious	(cheat,lie,deceive), (betrayed,untrust,faithless), (shout,yell,scream), (disrespect,insult,abuse), (so,quite,very), (insult,abuse,offend)
Sad	(lose,miss,lost), (upset,frustrated,disappointed), (failure,sorrow,regret), (unhappy,heartbroken,worried), (cry,weep,tears), (depressed,sentimental,unhappy)
Devastated	(die,pass,lose), (despair,grief,sadness), (lost,gone,missing), (crushed,shocked,stunned), (really,very,truly), (divorce,cancer,separation)
Excited	(thrilled, elated, happy), (eager, keen, enthusiastic), (wonder, awe, astonished), (party, fun, celebration), (happy, cheerful, delighted), (cheer, celebrate, shout)
Proud	(honor, dignity, respect), (accomplished, successful, celebrated) (award, recognition, medal), (achievement, accomplishment, success), (happy, cheerful, delighted), (pleased, satisfied, content)
Joyful	(happy, cheerful, delighted), (celebration, festivity, party) (fun, enjoyment, pleasure), (smile, laugh, enjoy), (glad, pleased, satisfied), (laughter, amusement, thrill)
Grateful	(thankful, appreciative, obliged), (thanks, appreciation, gratitude) (blessing, gift, favor), (kindness, support, help) (acknowledgment, recognition, appreciation),(smile, thank, express)

Table 5: List of top-6 most relevant semantic anchors to 10 emotion classes; each anchor is annotated by three words whose embeddings are closest to it.

B Details on Datasets and Pre-processing

(1) **Empathetic Dialogue** (Rashkin et al., 2019) consists of dialogues between a speaker and a listener with 32 single emotion label. For fair comparison with the previous model (Chen et al., 2023), we only utilize the first turn of the dialogue. The training/validation/test split of the dataset is 19,533 / 2,770 / 2,547, respectively.
(2) **GoEmotion** (Demszky et al., 2020) is a dataset of Reddit comments where each sample is annotated with one or more labels from 27 emotions and neutral. Following Chen et al. (2023), we exclude samples with multiple labels and the neutral label. The training/validation/test split of the remaining dataset is 23,485 / 2,956 / 2,984.
(3) **CancerEmo** (Sosea and Caragea, 2020) composes of 8500 sentences sampled from an online cancer survivors network and label them with 8 eight Plutchik basic emotions (Plutchik, 1980).
(4) **ISEAR** (Scherer and Wallbott, 1994) includes personal reports of emotional experiences from diverse cultural backgrounds. This collection comprises 7000 sentences, which are categorized into seven distinct emotions. The train/validation/test split of the dataset is 4,599 / 1,533 / 1,534.
(5) **GoEmotion-EK** (Ekman et al., 1999) annotates data originally constructed by (Demszky et al., 2020) into Ekman's 6 basic emotions. Following Yin and Shang (2022), sentences with multi labels and the neutral label are removed. The training/validation/test split of the remaining dataset is 23,485 / 2,956 / 2,984.
(6) **EmoInt** (Mohammad and Bravo-Marquez, 2017) comprises tweets of 4 emotion classes. The train/validation/test split of this dataset is 3,612 / 346 / 3,141.

	Empathetic Dialogue 32		GoEmotions 27		CE 8	IS 7	EK 6	EM 4
	Acc	Weighted F1	Acc	Weighted F1	Macro F1			
PLM-BiLSTM	54.3 ± 0.8	55.6 ± 0.7	62.3 ± 0.6	63.5 ± 0.6	72.5 ± 0.3	69.5 ± 0.7	71.5 ± 1.0	85.4 ± 0.4
PLM-DNN	52.9 ± 0.4	54.4 ± 0.5	62.0 ± 0.7	63.3 ± 1.3	73.0 ± 0.5	70.1 ± 0.7	71.9 ± 0.8	85.0 ± 0.2
PsyLing	56.0 ± 1.0	56.3 ± 0.9	62.7 ± 0.7	63.8 ± 1.1	74.4 ± 0.6	70.1 ± 1.0	71.0 ± 0.7	85.4 ± 0.5
KNNEC	57.1 ± 0.8	57.5 ± 0.8	63.6 ± 1.3	63.5 ± 1.0	73.9 ± 0.4	69.5 ± 0.5	72.7 ± 0.4	85.7 ± 0.4
Ours	**60.5 ± 0.3**	**61.0 ± 0.2**	**66.7 ± 0.4**	**66.4 ± 0.5**	**76.1 ± 0.3**	**72.2 ± 0.6**	**73.9 ± 0.5**	**86.5 ± 0.4**
Δ	+ 3.4%	+ 3.5%	+ 3.1%	+ 2.6%	+ 1.7%	+ 2.1%	+ 1.2%	+ 0.8%

Table 6: Classification results (in %) using $\text{BERT}_{\text{base}}$ as backbone for all methods, with weighted F1 and accuracy for fine-grained task and Macro F1 for coarse-grained task. The best/second-best results highlighted in **bold**/ underline. CE, IS, EK, EM stands for CancerEMO, ISEAR, GoEmotion-EK, EmoInt; numerals are the number of classes. Δ represents the improvement of our model over the second-best.

C Comparison with BERT-based model and Baseline settings

C.1 Comparison with BERT-based models

Some Baselines report official results utilizing $\text{BERT}_{\text{base}}$ as their backbone. For fair comparison, we incorporate our method on top of $\text{BERT}_{\text{base}}$, and the comparative results can be found in Table 6. We performed the experiment five times using different random seeds and reported the mean score along with the standard deviation.

C.2 Parameter settings of baseline models

For baseline (1-6), we uniformly set the batch size to 64, the learning rate to 2e-5, use AdamW as the optimizer, and set the weight decay to 0.01.

For baseline (7-12), We select parameters from the following range and determine their values based on performance on the validation set. These parameter candidates have subsumed their recommended parameters (if reported in their papers). The batch size is chosen from the set {4, 8, 16, 32, 64}, the learning rate from {1e-5, 2e-5, 1e-4, 1e-3, 1e-2}, the weight decay from {1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 0}, and the optimizer from Adam and AdamW.

D Details on 4 confusable subsets of ED

The 4 subsets of Empathetic Dialogue are selected by Suresh and Ong (2021), comprising the most challenging subsets identified after evaluating all possible combinations of four labels. These subsets include: a: {Anxious, Apprehensive, Afraid, Terrified}, b: {Devastated, Nostalgic, Sad, Sentimental}, c: {Angry, Ashamed, Furious, Guilty}, and d: {Anticipating, Excited, Hopeful, Guilty} from the Empathetic Dialogue datasets.

E Comparisons with LLMs on FEC tasks

Given the widespread application and promising outcomes of large language models, we further include GPT-4o and Llama3-8b, two highly popular and competitive LLMs in new comparisons on 2 largest fine-grained emotion classification datasets in the paper: Empathetic Dialogue and GoEmotions, using the popular experimental settings as Liu et al. (2024) and the prompt template used by Gao et al. (2023).

Experimental results are shown in Table 7, where ZS, FS denotes zero-shot and few-shot; ED, GE represents Empathetic Dialogue and GoEmotions respectively. The prompt template used for GoEmotions data is shown in Table 8.

	Empathetic Dialogue 32		GoEmotions 27	
	Acc	Weighted F1	Acc	Weighted F1
Llama3-8b-ZS	16.3 ± 0.5	11.5 ± 0.2	31.4 ± 0.4	28.1 ± 0.4
Llama3-8b-FS	18.9 ± 0.5	14.6 ± 0.3	31.6 ± 0.5	30.2 ± 0.1
GPT-4o-ZS	20.2 ± 0.4	19.2 ± 0.3	42.2 ± 0.2	42.7 ± 0.7
GPT-4o-FS	20.5 ± 0.3	20.2 ± 0.1	43.8 ± 0.3	44.0 ± 0.1
Ours	**61.2 ± 0.3**	**62.2 ± 0.2**	**67.6 ± 0.4**	**67.4 ± 0.5**

Table 7: Comparisons with GPT-4o and Llama3-8b on GE and ED datasets. The best results highlighted in **bold**.

As shown in Table 7, we can see that the two LLMs perform less satisfactorily in zero / few-shot experiments on these two difficult fine-grained emotion classification tasks. In fact, similar observations were also made by other researchers (Liu et al., 2024; Kocoń et al., 2023; Zhang et al., 2023). Indeed, why powerful LLMs do not excel in fine-grained emotion classification remains open and could be related to many factors: processing and understanding context correctly and extracting fine-grained structured sentiment (Kocoń et al., 2023), potential loss of structured emotional detail in the sentence (Liu et al., 2024; Zhang et al., 2023), etc.

*Prompt template for **zero-shot** emotion classification:*

Given sentences from Reddit comments, the task is to classify the sentences as being an *Admiration, Approval, Annoyance, Gratitude, Disapproval, Amusement, Curiosity, Love, Optimism, Disappointment, Joy, Realization, Anger, Sadness, Confusion, Caring, Excitement, Surprise, Disgust, Desire, Fear, Remorse, Embarrassment, Nervousness, Pride, Relief, or Grief* category of emotions. Don't explain yourself.

Thus given the following input:

input: [INPUT SENTENCE]
answer:

*Prompt template for **few-shot** emotion classification:*

Given sentences from Reddit comments, the task is to classify the sentences as being an *Admiration, Approval, Annoyance, Gratitude, Disapproval, Amusement, Curiosity, Love, Optimism, Disappointment, Joy, Realization, Anger, Sadness, Confusion, Caring, Excitement, Surprise, Disgust, Desire, Fear, Remorse, Embarrassment, Nervousness, Pride, Relief, or Grief* category of emotions. Don't explain yourself.

Some examples are:
input: [INPUT SENTENCE 1]
answer: [ANSWER 1]
input: [INPUT SENTENCE 2]
answer: [ANSWER 2]

Thus given the following input:
input: [INPUT SENTENCE]
answer:

Table 8: Prompt template used for GoEmotions data

We will study how to combine the advantages of specific classifiers with general LLMs for emotion classification in our future work.