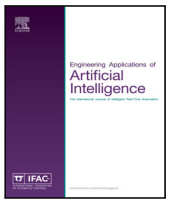
Engineering Applications of Artificial Intelligence 179 (2026) 115184

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Improving speech emotion recognition through self-attention and extremely randomized trees based weighted fuzzy concept-cognitive learning

Weihua Xu *, Kaiping Hu

College of Artificial Intelligence, Southwest University, Chongqing, 400715, PR China

ARTICLE INFO

Keywords:
Concept-cognitive learning
Granular computing
Human–computer interaction
Speech emotion recognition
Self-attention

ABSTRACT

Speech emotion recognition (SER) faces two recurring difficulties, namely the fuzzy nature of emotional expression and the limited transparency of mainstream classifiers. Concept-cognitive learning (CCL) offers an alternative classifier design in which decisions are made by matching samples to concepts in a hierarchical concept space. This study integrates CCL into an SER pipeline. A self-attention (SA) frontend refines acoustic low-level descriptors (LLDs) including mel-frequency cepstral coefficients (MFCC), log-mel spectrograms (LMS), zero-crossing rates (ZCR), chromagrams, and root mean square (RMS) energy. A weighted fuzzy CCL classifier with attribute weights supplied by extremely randomized trees (ERT) then maps the refined features into a fuzzy concept space where each utterance is matched to its closest progressive concept. The resulting model is denoted as ERT-WFCCL. On the EMO-DB dataset, the SA frontend contributes an improvement of 2.55 percentage points over a softmax classifier utilizing raw LLDs, the ERT-WFCCL classifier adds a further 0.38 percentage points, and the full pipeline reaches an accuracy of 81.11%. Two-tailed paired t-tests across four public benchmarks covering English, German, and Chinese, namely EMO-DB, RAVDESS, SAVEE, and CASIA, show that ERT-WFCCL outperforms nine of ten baselines on EMO-DB and remains competitive with the strongest neural and linear baselines on the other datasets. For each prediction, the classifier also records the matched concept and a ranked list of attribute contributions as a computational decision trace.

1. Introduction

Speech is a natural medium for conveying not only semantic meaning but also complex psychological states, making automatic speech emotion recognition (SER) a core component of empathetic human–computer interaction (de Lope and Graña, 2023; Hashem et al., 2023). Recognizing emotional expression from speech is a long-standing task that has been studied across several application domains (Liu et al., 2024).

The overall process of a speech emotion recognition system is illustrated in Fig. 1. After standardizing audio lengths through initial preprocessing, we extract features from both time and frequency domains. While common features include spectral (Cummins et al., 2015) and Teague energy operator-based types (Akçay and Oğuz, 2020), this study utilizes low-level descriptors (LLDs) (Ahmed et al., 2023) to capture short-term acoustic changes (Liu et al., 2024). We segment the speech signal into frames and extract ZCR, RMS, chromagram, LMS, and MFCC features to provide foundational time–frequency information reflecting emotional fluctuations.

Early SER systems relied on linear and statistical models, support vector machines (SVM) partitioned the feature space with hyperplanes (Luengo et al., 2005), while hidden Markov models and Gaussian mixture models captured the temporal dependence and non-stationary characteristics of speech signals (Nwe et al., 2003; Latif et al., 2021). Although effective, these methods depend heavily on hand-crafted features and struggle to model high-level emotional patterns. Deep learning methods, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory networks (LSTM) (Zhao et al., 2019; Li et al., 2021), overcame this limitation by automatically learning complex representations and improved recognition accuracy on a range of SER benchmarks. Their complex internal structures, however, lack decision trace, limiting their transparency in practical applications. Standard neural networks usually apply rigid decision boundaries. Human emotions are intrinsically fuzzy, and their acoustic features frequently overlap. Motivated by these two characteristics, we examine concept-cognitive learning (CCL) as an alternative classification framework for SER.

Concept-cognitive learning (CCL) is an emerging learning paradigm in artificial intelligence and cognitive computing (Guo et al., 2024; J. Li et al., 2017). Rather than imposing hard classification boundaries, CCL operates on a hierarchical concept space in which membership is graded. Each prediction can additionally be associated with the

* Corresponding author.
E-mail addresses: chxuwh@swu.edu.cn (W. Xu), hkp0922@email.swu.edu.cn (K. Hu).

https://doi.org/10.1016/j.engappai.2026.115184
Received 18 February 2025; Received in revised form 19 May 2026; Accepted 20 May 2026
Available online 28 May 2026

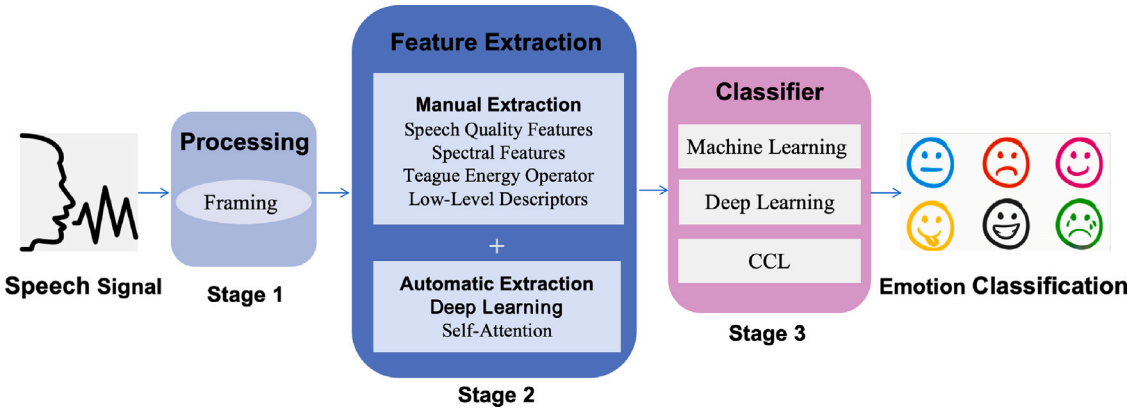


Fig. 1. The general workflow of SER tasks. The framework consists of three stages: preprocessing, feature extraction combining manual LLDs and SA-based deep features, and classification.

matched concept and a ranked list of attribute contributions, which provides a per-sample decision trace. CCL has been applied to other domains, including medical diagnosis (Guo and Xu, 2023). CCL has been extended along multiple directions. Early efforts explored fuzzy-based (Mi et al., 2020) and memory-based (Guo et al., 2023b) formulations to enrich concept representation. Subsequent work broadened the framework through incremental (Zhang et al., 2023) and semi-supervised (Mi et al., 2022) strategies, while two-way (Xu et al., 2023) and three-way (Guo et al., 2023a) models further refined the granularity of cognitive decisions. Among these, weighted fuzzy CCL is particularly relevant to this study because it allows flexible selection of weighted concepts to improve the applicability of concept analysis. A central challenge in this direction is how to determine attribute weights effectively. Existing studies have addressed this from complementary perspectives, including information entropy (Zhang et al., 2012), attribute weight correlation (Singh et al., 2017), and decision information granules (Zhang et al., 2023), each contributing to more robust weight computation. More recently, Xu and Zhang (2025) combined random forest (RF) with fuzzy CCL to optimize prediction results. Despite its advantages, RF requires an exhaustive search for optimal split points, which slows training and can weaken generalization. To overcome this, we adopt ERT (Geurts et al., 2006), which introduces greater randomness to reduce variance and provides a more efficient solution for fuzzy formal concept analysis.

In this study, we evaluate the proposed ERT-WFCCL model on four widely used public benchmark datasets: EMO-DB (Burkhardt et al., 2005), RAVDESS (Livingstone and Russo, 2018), SAVEE (Jackson and Haq, 2014), and CASIA (Y. Li et al., 2017). Acoustic features are extracted following Ahmed et al. (2023), refined by the self-attention front-end, and passed to the ERT-WFCCL classifier for emotion class assignment.

- The weighted fuzzy CCL framework is adapted to SER. The fuzzy concept space replaces a hard decision boundary with graded similarity to weighted concepts, and ERT-derived feature importance scores supply the attribute weights. This configuration is referred to as ERT-WFCCL.
- ERT-WFCCL is integrated with a self-attention frontend that refines acoustic LLDs. A controlled ablation on EMO-DB demonstrates that compared to a softmax on raw LLD baseline, the SA frontend contributes the larger share of the accuracy improvement by 2.55 percentage points, while the ERT-WFCCL classifier applied to SA features adds a smaller incremental margin of 0.38 percentage points. The framework therefore primarily operates by refining an already strong spectral representation.
- The integrated pipeline is evaluated on four public benchmarks including EMO-DB, RAVDESS, SAVEE, and CASIA covering three languages. Evaluation is conducted under stratified ten-fold and leave-one-speaker-out cross-validation using two-tailed paired t-tests against ten classical and deep baselines utilizing shared SA refined features. A decision trace consisting of the matched concept and the ranked attribute contributions is reported for each sample. This trace serves as a computational property of the classifier rather than a semantic or perceptual explanation.

The rest of this paper is organized as follows. Section 2 introduces the related concepts of fuzzy formal concept analysis (FFCA) and ERT. Section 3 discusses the cognitive learning process of weighted fuzzy concept space in detail, including how to classify the class label and dynamically update the weighted fuzzy concept space. Section 4 describes the dataset, the feature extraction method, and the implementation of the model classification. The experimental results and analyses are discussed in Section 5. The final section summarizes the study and outlines future research directions.

2. Preliminaries

This section introduces a novel method for learning weighted fuzzy concepts based on feature importance derived from an ensemble classifier.

2.1. Fuzzy formal concept analysis

In the framework of FFCA, the concept of a fuzzy formal context was first introduced by Yahia et al. (2000), and it serves as a mathematical instrument for both data analysis and knowledge representation. This context typically involves non-empty finite sets of objects and attributes, along with a fuzzy relation that connects the two sets, which is be crucial for the subsequent discussion. Let E represent a generic universe, and let $\tilde{F}$ be a fuzzy set defined over E, characterized by a membership function $\tilde{F}(\cdot) : E \to [0,1]$. For any element $e \in E$, the value $\tilde{F}(e)$ is the fuzzy membership degree of e within the fuzzy set $\tilde{F}$. The set of all fuzzy subsets of E is denoted as $\mathcal{H}(E)$.

Let $\tilde{F}_1$ and $\tilde{F}_2$ be two fuzzy sets defined over the universe E. If $\forall e \in E$, it holds that $\tilde{F}_1(e) \leq \tilde{F}_2(e)$, we say that $\tilde{F}_1$ is contained within $\tilde{F}_2$, denoted as $\tilde{F}_1 \subseteq \tilde{F}_2$. Additionally, we use $\mathcal{O}(E)$ to represent the collection of all crisp subsets of E.

A fuzzy formal context is represented by a triplet $(U, V, \tilde{S})$, where:

- $U = \{u_1, u_2, \ldots, u_n\}$ denotes the set of objects;
- $V = \{v_1, v_2, \ldots, v_m\}$ denotes the set of attributes;
- $\tilde{S} : U \times V \to [0,1]$ represents the fuzzy relation, indicating the degree of membership of object u_i to attribute v_j.

Consider a fuzzy formal context represented by $(U, V, \tilde{S})$, for any object subset $A \subseteq U$ and fuzzy attribute set $\tilde{B} \in \mathcal{H}(V)$, we define the following two operators $H : \mathcal{O}(U) \to \mathcal{H}(V)$ and $O : \mathcal{H}(V) \to \mathcal{O}(U)$:

$$H(A)(v_j) = \bigwedge_{u_i \in A} \tilde{S}(u_i, v_j), \quad \forall v_j \in V. \tag{1}$$

Table 1
A fuzzy formal decision context.

U	v_1	v_2	v_3	d
u_1	0.4	0.6	0.8	0
u_2	0.7	0.5	0.4	1
u_3	0.3	0.9	0.6	0
u_4	0.8	0.7	0.5	1

$$O(\tilde{B}) = \{u_i \in U | \forall v_j \in V, \tilde{B}(v_j) \leq \tilde{S}(u_i, v_j)\}. \quad (2)$$

A pair $(A, \tilde{B})$ forms a fuzzy concept if $H(A) = \tilde{B}$ and $O(\tilde{B}) = A$ where A is referred to as the extent and $\tilde{B}$ as the intent.

Let $(U, V, \tilde{S})$ and (U, D, T) represent two distinct fuzzy formal contexts, where:

- $\tilde{S} : U \times V \to [0, 1]$ denotes the fuzzy relation in the first context;
- $T : U \times D \to \{0, 1\}$ represents the crisp relation in the second context.

When the intersection of V and D is empty, the structure $(U, V, \tilde{S}, D, T)$ is referred to as a fuzzy formal decision context. In this context, V is the set of conditional attributes, while D corresponds to the set of decision attributes.

Example 1. In this fuzzy formal decision context as in Table 1, $U = \{u_1, u_2, u_3, u_4\}$ represents the object sets, while $V = \{v_1, v_2, v_3\}$ contains the attribute sets used for emotion recognition. Specifically, v_1 denotes the speech rate, v_2 refers to pitch, and v_3 indicates energy. The decision attribute d defines the emotion class, where d_1 corresponds to a happy emotion and d_2 represents a frustrated emotion. The fuzzy relationship $\tilde{S}$ is shown in Table 1, where $\tilde{S}(u_1, v_1) = 0.4$ means that the sample u_1 has a membership degree of 0.4 in v_1. The decision attribute d divides the set U into two classes: $D_1 = \{u_1, u_3\}$ for happy emotions, and $D_2 = \{u_2, u_4\}$ for frustrated emotions. The generated fuzzy concepts are summarized in Table 2, where the concept $(\{u_1, u_3\}, (\frac{0.3}{v_1}, \frac{0.6}{v_2}, \frac{0.6}{v_3}))$ is condensed as $(13, (0.3, 0.6, 0.6))$ for simplicity.

2.2. Extremely Randomized Trees

Extremely Randomized Trees (ERT) is an ensemble learning method, introduced by Geurts et al. (2006), that extends the concept of decision trees by introducing an additional layer of randomness. While conceptually similar to RF, Extremely Randomized Trees differ from RF by amplifying the degree of randomness in the tree-building process, particularly during node splitting. This greater degree of randomness helps reduce the variance of the model, thereby improving its ability to generalize to unseen data. Simultaneously, because the split points are chosen randomly, computational overhead is substantially reduced compared to traditional decision trees.

Feature importance analysis is a crucial tool for measuring the contribution of individual features to the predictive performance of a model. By analyzing the importance of features, it becomes possible to discern which features exert the most influence on the model's predictions, enabling not only the selection of the most relevant features but also simplifying the model and enhancing its decision trace. In the context of both Random Forests and Extremely Randomized Trees, feature importance is typically assessed through the cumulative reduction of the Gini index and the amount of information gain achieved by each feature.

The Gini index itself is a metric that gauges the "impurity" of a dataset, where higher values signify a more chaotic and less structured distribution of the data (Zhang et al., 2024). The Gini index for a node is derived from the distribution of class labels among the objects within that node q. The equation used to compute the Gini index is presented in Eq. (3).

Table 2
Fuzzy concepts of Table 1.

Concept	Concept name
$(U, (0.3, 0.5, 0.4))$	C_1
$(24, (0.7, 0.5, 0.4))$	C_2
$(13, (0.3, 0.6, 0.6))$	C_3
$(4, (0.8, 0.7, 0.5))$	C_4
$(3, (0.3, 0.9, 0.6))$	C_5
$(2, (0.7, 0.5, 0.4))$	C_6
$(1, (0.4, 0.6, 0.8))$	C_7
$(\emptyset, (1, 1, 1))$	C_8

$$G = 1 - \sum_{m=1}^{\gamma} p_m^2. \quad (3)$$

Here, γ represents the total number of categories, and p_m denotes the likelihood of an object belonging to class m. In both RF and ERT, each individual tree performs data splits based on distinct features, each of which contributes to lowering the overall Gini index, thereby improving the model's capacity to differentiate among categories. The importance of a feature is quantified by aggregating the total reduction in the Gini index attributable to that feature when it is used as a splitting criterion in all trees. In other words, the more a feature reduces the Gini index, the greater its contribution to the model's overall performance, and thus, the higher its importance. The following formula and steps outline the process for calculating feature importance in this context through Eq. (4).

$$VI(v_i) = \frac{1}{T} \sum_{t=1}^{T} \sum_{n \in N_t, v_n = v_i} P(n) \cdot \Delta G(v_i, n), \quad (4)$$

where $VI(v_i)$ denotes the importance score of feature v_i, and T denotes the overall count of trees within the forest. Each tree t consists of a set of nodes N_t, where a node n splits the data using a feature v_n. $P(n)$ represents the weight of node n (the proportion of samples in the node to the total number of samples). The term $\Delta G(v_i, n)$ captures the change in impurity induced by the split at node n, defined as

$$\Delta G(v_i, n) = G(n_{\text{parent}}) - \left(G(n_{\text{left}}) + G(n_{\text{right}})\right). \quad (5)$$

Here, $G(n_{\text{parent}})$, $G(n_{\text{left}})$, and $G(n_{\text{right}})$ denote the Gini values corresponding to the parent node, left child, and right child nodes individually. Common measures of impurity include the Gini index and entropy. To compute $VI(v_i)$, all nodes where $v_n = v_i$ are identified, and their respective $\Delta G(v_i, n)$ values are summed across all trees. The total contribution of feature v_i is then averaged over the total number of trees T, yielding the final importance score.

The resulting importance score $VI(v_i)$ is a non-negative value, with higher values indicating a greater contribution of the feature to reducing impurity. Importantly, the importance scores of all features in the model are normalized such that their sum equals 1. This framework effectively ranks features based on their ability to influence the model's decisions, providing a valuable decision trace tool for feature selection and model evaluation.

3. ERT-WFCCL

This section presents the ERT-WFCCL classifier, which is built from three components. An attribute weighting step first assigns ERT-derived importance scores to the input features, a construction step then builds a weighted fuzzy concept space from these weights, and an incremental update mechanism subsequently revises the concept space whenever new objects are introduced. The remainder of this section formalizes each component.

3.1. Weight calculation using feature importance

In a fuzzy formal decision context, denoted as $(U, V, \tilde{S}, D, T)$, $U = \{u_1, u_2, \ldots, u_n\}$ represents the objects, and $V = \{v_1, v_2, \ldots, v_m\}$ represents the attributes. The set of decision attributes is $D = \{d_1, d_2, \ldots, d_r\}$, and the decision partition of U with respect to D is given by $U/D = \{D_1, D_2, \ldots, D_t\}$, which divides U into distinct decision classes based on the decision attributes.

Definition 1. Let $(U, V, \tilde{S}, D, T)$ be a fuzzy formal decision context. The fuzzy relation $\tilde{S} : U \times V \to [0, 1]$ indicates the membership degree $\tilde{S}(u, v)$ of object $u \in U$ to attribute $v \in V$. Then the feature importance $VI(v)$ for each $v \in V$ is computed using the Eq. (4). The attribute v is assigned a weight, represented by $\omega(v)$, which is expressed as

$$\omega(v) = \mathrm{VI}(v), \tag{6}$$

where $\omega(v)$ satisfies $\sum_{v \in V} \omega(v) = 1$. The weight vector $\omega = (\omega(v_1), \omega(v_2), \ldots, \omega(v_m))$. For a fuzzy concept pair $(\mathcal{Z}, \tilde{\mathcal{B}})$, where $\mathcal{Z} \subseteq U$ denotes the extent and $\tilde{\mathcal{B}} \in \mathcal{H}(V)$ denotes the intent, the weight of this fuzzy concept $(\mathcal{Z}, \tilde{\mathcal{B}}, \omega)$ is expressed as follows:

$$\omega = \frac{1}{|V|} \sum_{v_i \in V} \tilde{\mathcal{B}}(v_i)\omega(v_i), \tag{7}$$

where $\tilde{\mathcal{B}}(v_i)$ represents the degree of fuzzy membership of the attribute v_i in the intent. The weight ω represents the average information importance of extent $\mathcal{Z}$, reflecting the significance of the multi-attribute intent in the fuzzy concept. Given two weighted fuzzy concepts $(\mathcal{Z}_1, \tilde{\mathcal{B}}_1, \omega_1)$ and $(\mathcal{Z}_2, \tilde{\mathcal{B}}_2, \omega_2)$, the hierarchical order relation is defined as

$$\begin{aligned}(\mathcal{Z}_1, \tilde{\mathcal{B}}_1, \omega_1) \le (\mathcal{Z}_2, \tilde{\mathcal{B}}_2, \omega_2) \\ \iff \mathcal{Z}_1 \subseteq \mathcal{Z}_2 \\ \iff \tilde{\mathcal{B}}_2 \le \tilde{\mathcal{B}}_1 (\text{or } \omega_2 \le \omega_1).\end{aligned} \tag{8}$$

This hierarchical order implies that the inclusion relation among fuzzy concepts depends on the subset relationship of extents, the membership degree relationship of intents, and the weight magnitudes. The weighted fuzzy concept lattice, denoted as $L_\omega(U, V, \tilde{S}, D, T)$, is formed by the set of all weighted fuzzy concepts.

The following proposition demonstrates the constructiveness of weighted fuzzy concepts using weights derived from ERT.

Proposition 1. *Consider a fuzzy formal decision context $(U, V, \tilde{S}, D, T)$. In which ω indicates the vector of weights associated with the attributes in V computed using the ERT. For any $\mathcal{Z}_1 \subseteq U$, the triple $(O(H(\mathcal{Z}_1)), H(\mathcal{Z}_1), \omega_1)$ is a weighted fuzzy concept.*

Proof. In accordance with Definition 1, for any $\mathcal{Z}_1 \subseteq U$, $(O(H(\mathcal{Z}_1)), H(\mathcal{Z}_1))$ is a fuzzy concept that satisfies $O(H(\mathcal{Z}_1)) = \mathcal{Z}_1$. The weight $\omega(v_i)$ is computed as the weighted sum of attribute importances, normalized over all attributes. Since $\omega(v_i)$ satisfies the normalization condition $\sum_{v_i \in V} \omega(v_i) = 1$, it follows that the triple $(O(H(\mathcal{Z}_1)), H(\mathcal{Z}_1), \omega_1)$ satisfies Definition 1.

Example 2. (Following Example 1) To explain the calculation process of weighted fuzzy concepts, we first analyze the attributes in the data and calculate their corresponding fuzzy weights based on each attribute in Table 1 and Eq. (7) as $\omega(v_1) = 0.4075$, $\omega(v_2) = 0.2350$, and $\omega(v_3) = 0.3575$. Subsequently, we list the weighted fuzzy concepts in Table 3. In Table 3, the intention of each weighted fuzzy concept node is represented by a set of assigned weights to quantify the importance and relative value of the node.

In cognitive processes, individuals typically prioritize certain nodes and selectively extract relevant information based on their preferences and needs. For instance, if a person is more inclined towards nodes with weights greater than 0.1819, then nodes such as $\mathcal{WC}_1$, $\mathcal{WC}_2$, $\mathcal{WC}_3$, and $\mathcal{WC}_6$ will be excluded, leaving only $\mathcal{WC}_4$, $\mathcal{WC}_5$, $\mathcal{WC}_7$, and $\mathcal{WC}_8$. This approach illustrates the adaptability of weighted fuzzy concepts, offering a more efficient method for solving human-centric problems and enhancing cognitive learning. By enabling the rapid gathering of relevant knowledge, this strategy significantly reduces the need for excessive storage space.

Table 3
The weighted fuzzy concepts obtained from Table 1.

Weighted fuzzy concepts	Concept name
(U, (0.3, 0.5, 0.4), 0.1276)	$\mathcal{WC}_1$
(24, (0.7, 0.5, 0.4), 0.1819)	$\mathcal{WC}_2$
(13, (0.3, 0.6, 0.6), 0.1592)	$\mathcal{WC}_3$
(4, (0.8, 0.7, 0.5), 0.2231)	$\mathcal{WC}_4$
(3, (0.3, 0.9, 0.6), 0.1828)	$\mathcal{WC}_5$
(2, (0.7, 0.5, 0.4), 0.1819)	$\mathcal{WC}_6$
(1, (0.4, 0.6, 0.8), 0.1967)	$\mathcal{WC}_7$
($\emptyset$, (1, 1, 1), 0.3333)	$\mathcal{WC}_8$

Table 4
A new fuzzy formal decision context.

U	v_1	v_2	d
u_1	0.12	0.54	0
u_2	0.15	0.63	0
u_3	0.08	0.49	0
u_4	0.25	0.72	0
u_5	0.69	0.42	1
u_6	0.82	0.51	1
u_7	0.67	0.29	1
u_8	0.79	0.35	1
u_9	0.27	0.65	0

From the preceding discussion, we derive the desired weighted fuzzy concepts. Notably, in Example 1, as the number of object dimensions increases, the exhaustive calculation of fuzzy concepts expands exponentially. Information granules, as a core concept in granular computing (Grc) theory, play a vital role in human cognition. To mitigate the computational complexity in cognitive learning, integrating information granules into the learning process becomes essential. Ultimately, the challenge lies in utilizing these weighted fuzzy granular concepts to construct a complete concept space, which remains a key area of focus in cognitive learning.

3.2. Generation of fuzzy concept space

In this subsection, we present a novel classification algorithm based on weighted fuzzy concepts and granular computing (Grc), aimed at constructing and refining concept spaces derived from weighted fuzzy granular concepts. The framework consists of two primary components: the establishment of an initial concept space to form the foundation for classification, and the continuous refinement of the concept space to adapt to data changes and meet classification requirements.

Consider $(U, V, \tilde{S}, D, T)$ as a fuzzy formal decision context, where $U/D = \{D_1, D_2, \ldots, D_i\}$ denotes the partition of the set of objects U based on the decision attributes D, and ω represents the vector of weights assigned to the attributes in V. For each subset D_i, the associated weighted fuzzy concept space C_i, is given by

$$C_i = \left\{ (\mathcal{O}(\mathcal{H}(v)), \mathcal{H}(v), \omega) \mid v \in D_i \right\}. \tag{9}$$

Furthermore, we denote $\mathcal{G} = \{C_1, C_2, \ldots, C_i\}$ as the weighted fuzzy concept space, where each C_i is considered a weighted fuzzy subspace of $\mathcal{G}$. It is crucial to emphasize that every object can be comprehensively learned, which aids in improving classification accuracy. Building on the prior explanation, we present the steps for constructing the weighted fuzzy concept space in Algorithm 1.

Algorithm 1: Constructing Weighted Fuzzy Concept Space

Input: A fuzzy formal decision context $(U, V, \tilde{S}, D, T)$.
Output: The weighted fuzzy concept space $\mathcal{G}$.

```
for D_i ∈ U/D do
    Initialize D_i ← ∅;
    for every v ∈ D_i do
        Calculate the weight value ω for the multi-attribute;
        Retrieve the weighted fuzzy concept (O(H(v)), H(v), ω);
        Add this concept to the corresponding subspace:
          C_i ← C_i ∪ {(O(H(v)), H(v), ω)};
    Append the subspace to the overall fuzzy concept space:
      G ← G ∪ {C_i};
return G = {C_1, C_2, ..., C_i}
```

Example 3. Table 4 outlines a novel fuzzy formal decision context $(U, V, \tilde{S}, D, T)$, where the eight objects are classified into two distinct groups according to the decision attribute d, namely $D_1 = \{u_1, u_2, u_3, u_4\}$ and $D_2 = \{u_5, u_6, u_7, u_8\}$. From Eq. (7), the attribute weight is given by $\omega = (0.73475, 0.26525)$. Consequently, the weighted fuzzy concept subspace C_1 associated with class D_1 is expressed as

$$C_1 = \begin{cases} \big(\{u_1\}, (0.12, 0.54), 0.0969\big), \\ \big(\{u_1, u_2, u_3\}, (0.08, 0.49), 0.0617\big). \end{cases}$$

Similarly, the weighted fuzzy concept subspace C_2 corresponding to class D_2 is given by:

$$C_2 = \begin{cases} \big(\{u_6\}, (0.82, 0.51), 0.4352\big), \\ \big(\{u_5, u_6\}, (0.69, 0.42), 0.3429\big), \\ \big(\{u_6, u_8\}, (0.79, 0.35), 0.3709\big), \\ \big(\{u_5, u_6, u_7, u_8\}, (0.67, 0.29), 0.2929\big). \end{cases}$$

The relationship between two weighted fuzzy concepts within C_i is not only influenced by the decision class D_i, but also by data noise. Such noise can diminish the strength of the correlation, making it essential to filter out the weighted fuzzy concepts affected by it. To tackle this challenge, we define a similarity measure, which is presented below.

Definition 2. Consider a fuzzy formal decision context $(U, V, \tilde{S}, D, T)$, where ω represents the weight vector associated with the attributes. Given a fuzzy subspace after weighting C_i and a threshold ε. If $(U_1, \tilde{V}_1, \omega_1)$ denotes a weighted fuzzy concept within C_i, and $(U_2, \tilde{V}_2, \omega_2)$ is its subconcept, the similarity between these two weighted fuzzy concepts is defined by the following equation:

$$\eta_{1,2}^{C_i} = \frac{|U_1 \cap U_2|}{|U_1 \cap U_2| + 2\left(\lambda|U_1 - U_2| + (1-\lambda)|U_2 - U_1|\right)}, \tag{10}$$

where $\lambda = |\omega_1 - \omega_2|$, and $|U_1|$ represents the cardinality of U_1 with respect to the concept $(U_1, \tilde{V}_1, \omega_1)$. Here, ω_1 and ω_2 denote the concept-level weights computed via Eq. (7), not the shared attribute weight vector ω. Since different concepts possess distinct intents $\tilde{V}_k$, their concept weights ω_k vary accordingly, and $\lambda = |\omega_1 - \omega_2|$ captures the difference in information importance between concept pairs.

Since $(U_2, \tilde{V}_2, \omega_2)$ is a subconcept of $(U_1, \tilde{V}_1, \omega_1)$, we know that $|U_2 - U_1| = 0$, so the equation can be simplified to

$$\eta_{1,2}^{C_i} = \frac{|U_1 \cap U_2|}{|U_1 \cap U_2| + 2\lambda|U_1 - U_2|}, \tag{11}$$

where $\eta_{1,2}^{C_i}$ quantifies the degree of similarity between $(U_1, \tilde{V}_1, \omega_1)$ and $(U_2, \tilde{V}_2, \omega_2)$. A larger value of $\eta_{1,2}^{C_i}$ indicates a stronger similarity. When $\eta_{1,2}^{C_i} > \varepsilon$, the prominence of the two fuzzy concepts with weights increases. On the other hand, when $\eta_{1,2}^{C_i} \le \varepsilon$, their importance decreases.

In practice, a weak connection between two concepts with weighted fuzzy relations may arise due to noise within C_i. In such cases, $(U_2, \tilde{V}_2, \omega_2)$ should be excluded from the construction of the weighted fuzzy concept space.

From the above, it is clear that the threshold ε plays a crucial role in controlling the scale of the concept space assigned weights. As ε increases, the resulting space becomes smaller. The procedure for updating this space is described in Algorithm 2.

Algorithm 2: Updating the Fuzzy Concept Space Assigned Weights

Input: An initial weighted fuzzy concept space $\mathcal{G}$ and a threshold ε.
Output: An updated weighted fuzzy concept space $\mathcal{G}^{\varepsilon}$.

```
for C_i ∈ G do
    for (O(H(v_t)), H(v_t), ω_t) ∈ C_i do
        Initialize C^ε_{i,t} ← ∅;
        if (O(H(v_j)), H(v_j), ω_j) is a fuzzy subconcept of
          (O(H(v_t)), H(v_t), ω_t) within C_i then
            Compute η^{C_i}_{t,j} as defined in Definition 2;
            if η^{C_i}_{t,j} > ε then
                Add (O(H(v_j)), H(v_j), ω_j) to C^ε_{i,t};
        C^ε_i ← ⋃_{O(H(v_t))∈C_i} C^ε_{i,t};
    Set G^ε ← C^ε_i;
return G^ε = {C^ε_1, C^ε_2, ..., C^ε_t}
```

Example 4. Given a threshold of $\varepsilon = 0.34$, The similarity of the fuzzy concepts involving weights for C_1 is calculated by: $\eta_{1,2}^{C_1} = 0.341 > 0.34$. Thus, C_1 is updated to C_1^{ε}, which is represented as

$$C_1^{\varepsilon} = \begin{cases} \big(\{u_1, u_2, u_3\}, (0.08, 0.49), 0.0617\big), \\ \big(\{u_1\}, (0.12, 0.54), 0.0969\big). \end{cases}$$

Similarly, the weighted fuzzy concept similarity for C_2 is calculated as follows: $\eta_{1,2}^{C_2} = 0.372 > 0.34, \eta_{1,3}^{C_2} = 0.374 > 0.34, \eta_{1,4}^{C_2} = 0.185 < 0.34, \eta_{2,3}^{C_2} = 0.333 < 0.34$.

Therefore, C_2 is updated to C_2^{ε} as

$$C_2^{\varepsilon} = \begin{cases} \big(\{u_6\}, (0.82, 0.51), 0.4352\big), \\ \big(\{u_5, u_6\}, (0.69, 0.42), 0.3429\big), \\ \big(\{u_6, u_8\}, (0.79, 0.35), 0.3709\big). \end{cases}$$

Definition 3. Let $(U, V, \tilde{S}, D, T)$ be a fuzzy formal decision context, where ω representing the attribute weights. Assume that a weighted fuzzy subspace C_{ε}^{i} contains fuzzy concepts $(U_1, \tilde{V}_1, \omega_1), (U_2, \tilde{V}_2, \omega_2), \ldots, (U_t, \tilde{V}_t, \omega_t)$, satisfying the hierarchical relation $U_1 \subseteq U_2 \subseteq \cdots \subseteq U_t$. Under these conditions, $(U_t, \tilde{V}_t, \omega_t)$ is defined as the supremum concept. The evolving fuzzy concept with weights is then constructed in the following form:

$$U_{i,j}^{\varepsilon} = U_1 \cup U_2 \cup \cdots \cup U_t, \tag{12}$$

$$\tilde{V}_{i,j}^{\varepsilon} = \frac{1}{2^{t-1}}\left(\tilde{V}_1 + \tilde{V}_2 + 2\tilde{V}_3 + 4\tilde{V}_4 + \cdots + 2^{t-2}\tilde{V}_t\right). \tag{13}$$

The base-2 progressive weighting scheme assigns proportionally larger weights to larger extents while the normalization factor $\frac{1}{2^{t-1}}$ keeps the computation in closed form.

Accordingly, $(U_{i,j}^{\varepsilon}, \tilde{V}_{i,j}^{\varepsilon}, \omega_{i,j}^{\varepsilon})$ represents a progressive weighted fuzzy concept, where $\omega_{i,j}^{\varepsilon}$ is computed as

$$\omega_{i,j}^{\varepsilon} = \frac{1}{|U|}\sum_{x_i \in X} \tilde{V}_{i,j}^{\varepsilon}(x_i)\omega(x_i). \tag{14}$$

The overall space of evolving fuzzy concept with weights, denoted by $\mathcal{M}^{\varepsilon}$, can be described as a collection of subspaces $\{M_1^{\varepsilon}, M_2^{\varepsilon}, \ldots, M_s^{\varepsilon}\}$, where each subspace is defined by

$$M_i^{\varepsilon} = \{M_{i,j}^{\varepsilon} \mid j = 1, 2, \ldots, m\} = \{(U_{i,j}^{\varepsilon}, \tilde{V}_{i,j}^{\varepsilon}, \omega_{i,j}^{\varepsilon})\}. \tag{15}$$

Here, m represents the total count of fuzzy concepts contained in the subspace C_i. The intent $\tilde{V}_{i,j}^{\varepsilon}$ quantifies the size of a progressive weighted fuzzy concept, where the weights assigned to intents vary based on their extents. Specifically, the extent U_i has a direct influence on the weight assigned to its corresponding intent $\tilde{V}_i$, with larger extents contributing to greater weights. The total weight of all intents is normalized to 1. Lastly, the procedure to compute the progressive weighted fuzzy concept space $\mathcal{M}^{\varepsilon}$ is detailed in Algorithm 3.

Algorithm 3: Constructing the Evolving Fuzzy Concept Space With Weights

Input: An updated fuzzy concept space with weights$\mathcal{C}^{\varepsilon} = \{C_1^{\varepsilon}, C_2^{\varepsilon}, \ldots, C_t^{\varepsilon}\}$ and a threshold ε.
Output: The evolving fuzzy concept space with weights $\mathcal{M}^{\varepsilon} = \{M_1^{\varepsilon}, M_2^{\varepsilon}, \ldots, M_t^{\varepsilon}\}$.

for $C_i^{\varepsilon} \in \mathcal{C}^{\varepsilon}$ **do**
  Initialize $Q_i, \hat{Q}_i, \hat{S}_i \leftarrow \emptyset$;
  for $(\mathcal{O}(\mathcal{H}(v_j)), \mathcal{H}(v_j), \omega_j) \in C_i^{\varepsilon}$ **do**
    for $(\mathcal{O}(\mathcal{H}(v_k)), \mathcal{H}(v_k), \omega_k) \in C_i^{\varepsilon}$ **do**
      Set $T_{i,j} \leftarrow \emptyset$;
      if $(\mathcal{O}(\mathcal{H}(v_k)), \mathcal{H}(v_k), \omega_k)$ *is a sub-concept of* $(\mathcal{O}(\mathcal{H}(v_j)), \mathcal{H}(v_j), \omega_j)$ **then**
        Update Q_i and $T_{i,j}$ by adding $(\mathcal{O}(\mathcal{H}(v_j)), \mathcal{H}(v_j), \omega_j)$ and $(\mathcal{O}(\mathcal{H}(v_k)), \mathcal{H}(v_k), \omega_k)$ respectively;
  Set $\hat{T}_i \leftarrow T_{i,j}$;
  for $T_{i,n} \in \hat{T}_i$ **do**
    if *there exists exactly one concept* $(\mathcal{O}(\mathcal{H}(v_t)), \mathcal{H}(v_t), \omega_t)$ *in* Q_i *such that all concepts in* $S_{i,n}$ *are sub-concepts of* $(\mathcal{O}(\mathcal{H}(v_t)), \mathcal{H}(v_t), \omega_t)$ **then**
      Set $S_{i,n}$ and $\hat{Q}_i \leftarrow \{(\mathcal{O}(\mathcal{H}(v_t)), \mathcal{H}(v_t), \omega_t)\}$;
  Compute the evolving weighted fuzzy concept $(E_{i,n}^{\varepsilon}, \tilde{F}_{i,n}^{\varepsilon}, \omega_{i,n}^{\varepsilon})$ as defined in Definition 3;
  Set $M_i^{\varepsilon} \leftarrow (E_{i,n}^{\varepsilon}, \tilde{F}_{i,n}^{\varepsilon}, \omega_{i,n}^{\varepsilon})$;
return $\mathcal{M}^{\varepsilon} = \{M_1^{\varepsilon}, M_2^{\varepsilon}, \ldots, M_t^{\varepsilon}\}$

Example 5. According to Definition 3, the progressive weighted fuzzy concepts are as follows:

$M_{1,1}^{\varepsilon} = \left(\{x_1, x_2, x_3\}, (0.14, 0.76), 0.07136\right)$;
$M_{2,1}^{\varepsilon} = \left(\{x_5, x_6, x_5, x_8\}, (1.125, 0.495), 0.6357\right)$.

Each decision class is associated with a single progressive weighted fuzzy concept. These concepts not only preserve the crucial information but also eliminate unnecessary weighted fuzzy concepts, thus optimizing the efficiency of the CCL process.

3.3. Evolving weighted fuzzy concept-based incremental cognitive learning

In a fuzzy formal decision context $(U, V, \tilde{S}, D, T)$, the progressive weighted fuzzy concept space demonstrates strong performance in data classification. Determining the class label for a newly introduced object Δu presents a significant challenge that merits further exploration. Furthermore, incorporating Δu modifies the evolving fuzzy concept space with weights derived from the initial context. This subsection describes the incremental learning mechanism used to build the evolving fuzzy concept space with weights.

Algorithm 4: Class Label Prediction

Input: The updated fuzzy concept space $\mathcal{M}^{\varepsilon} = \{M_1^{\varepsilon}, M_2^{\varepsilon}, \ldots, M_t^{\varepsilon}\}$ and new object Δu.
Output: Predicted class of Δu.

for $M_{\delta_i} \in \mathcal{M}^{\varepsilon}$ **do**
  for $M_{i,l}^{\varepsilon} \in M_i^{\varepsilon}$ **do**
    Compute $d(\Delta u, M_{i,l}^{\varepsilon})$ from Definition 4;
  Find the minimum distance: $md_i = \min(d(\Delta u, M_{i,l}^{\varepsilon}))$;
Identify the index $i^* = \arg\min_{i \in \{1,2,\ldots,t\}} md_i$, where md_i is the minimal distance for the i-th concept in $\mathcal{M}^{\varepsilon}$;
return *Predicted label of* Δu

3.3.1. Estimating class labels following the inclusion of new objects

Within a concept space, one can assess the similarity between objects by calculating the Euclidean distance based on their attribute characteristics. Simply put, a smaller distance corresponds to a greater similarity between the objects. Therefore, to label the new object Δu, it is essential to calculate the distance between Δu and the fuzzy concept with weights in $\mathcal{M}^{\varepsilon}$.

Definition 4. Consider a fuzzy formal decision context $(U, V, \tilde{S}, D, T)$, where ω denotes a vector of attribute weights. For an object Δu newly introduced into the system, with a membership value $\tilde{U}$ relative to $\tilde{S}$, the Euclidean distance calculated from Δu to the lth evolving concept $(U_{i,l}^{\varepsilon}, \tilde{V}_{i,l}^{\varepsilon}, \omega_{i,l}^{\varepsilon})$ within M_i^{ε} is expressed as

$$d(\Delta u, M_{i,l}^{\varepsilon}) = \sqrt{\sum_{b \in M} \left(\omega(b)\left(\tilde{V}(b) - \tilde{V}_{i,l}^{\varepsilon}(b)\right)\right)^2}. \tag{16}$$

The value of $d(\Delta u, M_{i,l}^{\varepsilon})$ measures how similar Δu is to the concept $(U_{i,l}^{\varepsilon}, \tilde{V}_{i,l}^{\varepsilon}, \omega_{i,l}^{\varepsilon})$. A smaller value of this distance indicates a stronger similarity, whereas a larger value reflects a weaker similarity. Classification of Δu follows the principle of selecting the concept with the smallest distance. If several distinct concepts share an identical minimum distance, Δu is classified based on a recognition priority rule.

Example 6. Within the fuzzy formal decision framework illustrated in Table 4, objects u_1 through u_8 are taken from Example 2, while u_9 is a newly introduced object. For the object u_9, the membership values with respect to $\tilde{S}$ are given as $\tilde{V} = (0.27, 0.65)$, and its true label is 0. The following step focuses on determining the Euclidean distance from u_9 to the present evolving fuzzy concept space $\mathcal{M}^{\varepsilon}$ with weights, outlined below

$$d(u_9, M_{1,1}^{\varepsilon}) = 0.0121, \quad d(u_9, M_{2,1}^{\varepsilon}) = 0.553.$$

Here, the smallest distance occurs between u_9 and $M_{1,1}^{\varepsilon}$ within $\mathcal{M}^{\varepsilon}$, indicating that u_9 should be assigned to decision class D_1, which aligns with its true label of 0.

The subsequent section introduces Algorithm 4, which elaborates on the method for determining class labels upon the random addition of new objects.

3.3.2. Dynamic update approach for the evolving weighted Fuzzy conceptual space

Upon the addition of a new element Δu, the first step involves predicting its class label using Algorithm 4. Let the predicted label be c, which leads to $U_c' = U_c \cup \Delta u$. To avoid recomputing all the weighted fuzzy concepts within U_c', Algorithm 5 presents a method for dynamically updating the evolving fuzzy concept space with weights.

The core idea behind this approach, especially in steps 3–9, assumes that the class label of Δu is c. Initially, the weight vector is updated

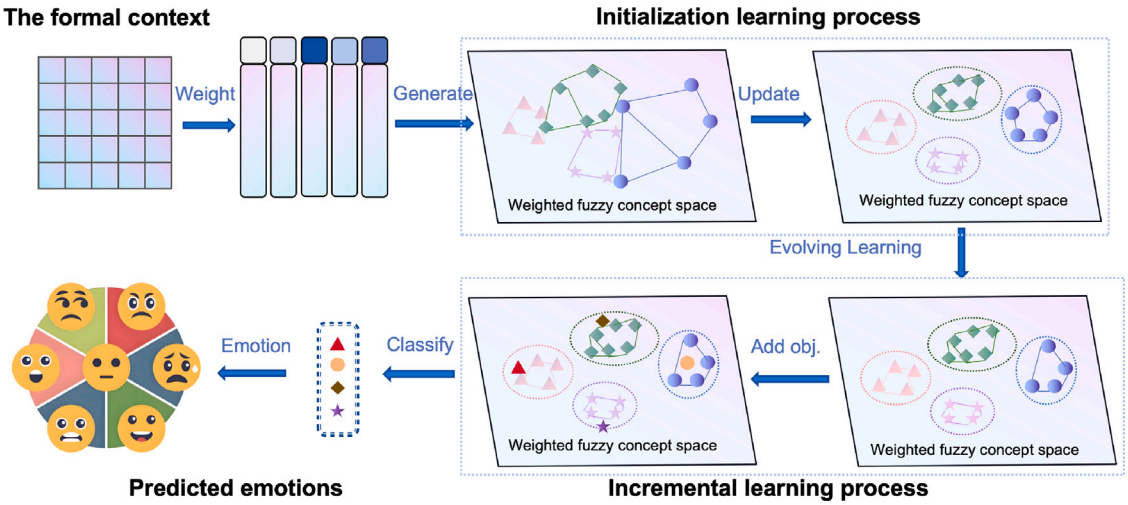


Fig. 2. The process of constructing and updating the fuzzy concept space with weights.

Algorithm 5: Dynamic Update of Evolving Weighted Fuzzy Concepts with New Object Addition

Input: The fuzzy concept space with weights $\mathcal{G} = \{C_1, C_2, \ldots, C_t\}$, the evolving fuzzy concept space with weights $\mathcal{M}^\varepsilon = \{M_1^\varepsilon, M_2^\varepsilon, \ldots, M_t^\varepsilon\}$, the decision partition $U/D = \{D_1, D_2, \ldots, D_t\}$, the newly added object εu, and the threshold ε.

Output: The newly updated evolving fuzzy concept space with weights $\hat{\mathcal{M}}^\varepsilon = \{\hat{M}_1^\varepsilon, \hat{M}_2^\varepsilon, \ldots, \hat{M}_t^\varepsilon\}$.

for εu **do**
- Compute the membership degree of εu to $\tilde{S}$, denoted as $\tilde{A}$;
- Use Algorithm 4 to determine the class label of εu, which is c;
- Update the weight vector ω and adjust the multi-attribute intent $\tilde{A}$ in M_a^ε $(a \neq c)$;
- Store the updated concept in $\hat{M}_a^\varepsilon$;
- Set $\hat{\mathcal{M}}^\varepsilon \leftarrow \hat{M}_a^\varepsilon$;
- Set $\hat{C}_c \leftarrow \emptyset$;
- **for** $u_c \in D_c$ **do**
  - **if** $\tilde{S}(\varepsilon u, v) \geq \tilde{S}(u_c, v)$ *for all* $v \in V$ **then**
    - Set $u_c^{\star\star} \leftarrow \varepsilon u$ and update $\hat{C}_c \leftarrow (u_c^{\star\star}, u_c^\star, \omega_c)$;
- Update $\hat{C}_j$ by computing $(\varepsilon u^{\star\star}, \varepsilon u^\star, \omega)$;
- Update the fuzzy concept space, resulting in $\hat{C}_c^\varepsilon$ using Algorithm 2;
- The updated evolving fuzzy concept space with weights is denoted as $\hat{\mathcal{M}}_c^\varepsilon$, as defined in Definition 4;

return $\hat{\mathcal{M}}^\varepsilon = \{\hat{M}_1^\varepsilon, \hat{M}_2^\varepsilon, \ldots, \hat{M}_t^\varepsilon\}$

based on the newly introduced data. Following this, the recalculation of the attribute-based weight values for intents in M_a^ε (with $a \neq c$) takes place. Subsequently, it becomes clear that M_a^ε is transformed into $\hat{M}_a^\varepsilon$. Let $|U|$ and $|V|$ denote the sizes of the object and attribute sets, respectively, and let $|D|$ represent the number of class labels, while $|D_a|$ corresponds to the size of the classification set within the decision partition. In this case, the computational complexity in the third step is given by: $O\left(\sum_{a=1}^{|D|} |D|(|D_a|^2 + |M_a^\varepsilon|)\right)$.

The primary distinction between the dynamic and static updating algorithms lies in how the newly generated weighted fuzzy concept area $\hat{C}_c$ is constructed. In the process of updating the fuzzy concept with weights for class a, the difference between Δu and x_c is first calculated, requiring $O(|V|)$ time. Afterward, computing the extent of the updated fuzzy concept is carried out in $O(|V|(|U| - u_c^{\star\star}))$. Therefore, the time complexity for steps 4–9 can be written as: $O\left(\sum_{c=1}^{|D_c|} |V|\left(1 + (|U| - u_c^{\star\star})\right)\right)$. In contrast, the static updating approach necessitates a full recalculation of all fuzzy concepts with weights, resulting in a time cost of: $O\left(|V|(1 + |U|^2)\right)$. To summarize, the overall time complexity for Algorithm 5 can be expressed as:

$$O\left(\sum_{a=1}^{|D|} |D|\left(|\mathcal{D}_a|^2 + |M_a^\varepsilon|\right) + \sum_{c=1}^{|D_c|} |V|\left(1 + (|U| - u_c^{\star\star})\right)\right).$$

From the analysis above, it is clear that the dynamic updating approach for the evolving weighted fuzzy concept space not only improves learning efficiency but also reduces the number of unnecessary iterations, as compared to the static updating method.

The effective integration of new objects with the foundational evolving fuzzy concept space with weights is a critical challenge in incremental learning, which warrants further exploration. Initially, the label of each newly introduced unlabeled data point is determined, after which the evolving fuzzy concept space is updated for incremental learning. This procedure utilizes a mechanism for dynamic updates based on concept learning, which effectively leverages the information granularity of the newly introduced object, enhancing classification under uncertainty and greatly improving classification efficiency. Algorithm 6 introduces a mechanism for dynamically updating the multi-object evolving fuzzy concept space with weights.

Algorithm 6: Adaptive Update of Evolving Weighted Fuzzy Concepts with the Addition of Multiple Objects

Input: The weighted fuzzy concept space $\mathcal{G} = \{C_1, C_2, \ldots, C_t\}$, the evolving fuzzy concept space with weights $\mathcal{M}^\varepsilon = \{M_1^\varepsilon, M_2^\varepsilon, \ldots, M_t^\varepsilon\}$, the decision partition $U/D = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_t\}$, the newly added object set $A = \{A_1, A_2, \ldots, A_k\}$, and threshold ε.

Output: The updated evolving weighted fuzzy concept space $\hat{\mathcal{M}}^\varepsilon = \{\hat{M}_1^\varepsilon, \hat{M}_2^\varepsilon, \ldots, \hat{M}_t^\varepsilon\}$ along with the class labels of the added objects.

for $A_i \in A$ **do**
- **for** $a_j \in A_i$ **do**
  - Determine the class label $c_{i,j}$ of a_j using Algorithm 4, and set $C(i, j) = c_{i,j}$;
  - Apply Algorithm 5 for updating the evolving weighted fuzzy concept space, resulting in $\hat{\mathcal{M}}^\varepsilon = \{\hat{M}_1^\varepsilon, \hat{M}_2^\varepsilon, \ldots, \hat{M}_t^\varepsilon\}$;

return *The class labels of* A *and the updated space* $\hat{\mathcal{M}}^\varepsilon$

As illustrated in Fig. 2, starting from a fuzzy formal decision context that includes a set of decision attributes, both the fuzzy concept

Table 5
Introduction to the speech datasets used in this study.

Dataset	Language	Speakers	Total samples	Emotions	Ref.	Download
EMO-DB	German	10 (5M, 5F)	535	7	Burkhardt et al. (2005)	Download
RAVDESS	English	24 (12M, 12F)	1440	8	Livingstone and Russo (2018)	Download
SAVEE	English	4 (Male)	480	7	Jackson and Haq (2014)	Download
CASIA	Chinese	4 (2M, 2F)	1200	6	Y. Li et al. (2017)	Download

subspace with assigned weights and the traditional decision concept space can be generated. To further refine the classification process, the updated fuzzy concept space, which incorporates importance factors, is introduced based on Definition 3. Each subspace, characterized by specific weights, is then linked to a distinct class label. In alignment with human cognitive processes, a refined fuzzy concept, adjusted for intensity, is created to remove any redundant information. For classifying multiple objects, their respective labels are identified within the newly constructed fuzzy concept space, which is organized according to priority. Consequently, the incremental learning strategy enhances classification precision by systematically integrating fresh data.

4. Experimental analysis

The SER framework is composed of two primary elements. One component is the preprocessing module, responsible for extracting relevant features from the speech data in the dataset. The other component is the classifier, which applies these features to perform the speech emotion recognition task. In this section, we describe in detail the benchmark datasets used, the specific process of feature extraction, the process of training the model and the specific implementation parameters set to ensure the reproducibility of the experimental results.

4.1. Public datasets of speech emotion

To ensure reproducibility and comprehensively validate the proposed model, we briefly describe the benchmark datasets used in this study. For a more nuanced evaluation of the proposed model, four datasets are used in this study: EMO-DB, SAVEE, RAVDESS, and CASIA covering three languages: English, German, and Chinese. These datasets are briefly described in the following.

4.1.1. The Berlin Emotional Speech Database (EMO-DB)

The Berlin Emotional Speech Database (Burkhardt et al., 2005) is widely recognized as the most prominent and extensively employed dataset within the research community focusing on Speech Emotion Recognition (SER). Speech samples are recorded at a frequency of 16 kHz with a precision of 16-bit depth. The database contains 535 recordings in German, which are categorized into seven distinct emotion types: “anger”, “fear”, “sadness”, “happiness”, “disgust”, “boredom” and “neutral” (Ahmed et al., 2023). However, this dataset exhibits class imbalance, as anger samples significantly outnumber other emotional classes.

4.1.2. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

RAVDESS (Livingstone and Russo, 2018) is a widely used dataset for SER tasks. It comprises English sentences performed by 12 male and 12 female actors, showcasing eight distinct emotional expressions through audio and video recordings. In this study, only audio samples were used. The dataset contains 1440 audio files with a sampling rate of 48 kHz, with each actor participating in 60 trials. The sample employed in this study encompasses the following eight emotion categories: “sad”, “happy”, “angry”, “calm”, “fear”, “surprise”, “disgust” and “neutral”. While the dataset is generally balanced, the “neutral” category contains slightly fewer samples than other categories.

4.1.3. The Surrey Audio-Visual Expressive Emotion Dataset (SAVEE)

The SAVEE dataset (Jackson and Haq, 2014) contains 480 recordings from four British male actors aged 27–31, spanning seven emotion categories: “anger”, “happiness”, “neutrality”, “disgust”, “sadness”, “fear” and “surprise”. All recordings were captured at 44.1 kHz with 16-bit precision. However, this dataset exhibits class imbalance, as the “neutral” category contains nearly twice the samples of other categories. Only speech audio samples were used in this study.

4.1.4. The CASIA Chinese Emotion Corpus (CASIA)

The CASIA dataset (Y. Li et al., 2017), created by the Institute of Automation, Chinese Academy of Sciences, includes 1200 Chinese speech examples. Each recording was captured at 16 kHz and encoded with 16-bit precision. The dataset covers 6 emotion categories: “anger”, “fear”, “happiness”, “calm”, “sadness”, and “surprise”. Each of the four actors recited 300 sentences from the same text and 100 sentences from different texts. The acquired speech signal is almost noise-free and of high quality. This dataset does not suffer from the class imbalance problem and there are 200 samples for each sentiment category. Table 5 shows an introduction to each speech dataset and the official download link.

4.2. Feature extraction

The effectiveness of SER systems heavily relies on the richness and accuracy of emotion-related characteristics derived from audio signals. High-quality features can reveal deep emotional patterns in speech, thereby helping the model to achieve more accurate emotion classification. The hand-crafted features contain rich emotional information, including both local instantaneous features and global vocal characteristics. In this study, we first extracted five different spectral features as input to the SER model. These features include mel-frequency cepstral coefficients (MFCC), log-mel spectrogram (LMS), zero crossing rate (ZCR), chromagram, and root mean square values (RMS).

After the initial feature extraction, we further integrate the self-attention mechanism to refine these features. By leveraging this mechanism, we can efficiently capture the relationships and dependencies between the features. This process not only improves the expression ability of features, but also mines the potential global context information, which provides more accurate input for sentiment classification. The self-attention mechanism enables the model to dynamically adjust the weight between different features, highlighting the most discriminative signals in emotion variations and ultimately optimizing classification accuracy. The following presents a brief summary of the features extracted.

4.2.1. Mel-frequency cepstral coefficients (MFCC)

MFCC is ubiquitous in speech and audio technology, and is used in applications such as automatic speech recognition, language identification (Joysingh et al., 2025). Research has shown that the human ear has varying sensitivity to different frequencies, especially within the 200 Hz to 5000 Hz range, which is crucial for speech clarity. Specifically, when there is a significant difference in the loudness of two frequency components, the louder component tends to mask the quieter one, a phenomenon known as the “masking effect.” The advantage of MFCC lies in its independence from the specific nature of the input signal, without making additional assumptions or restrictions.

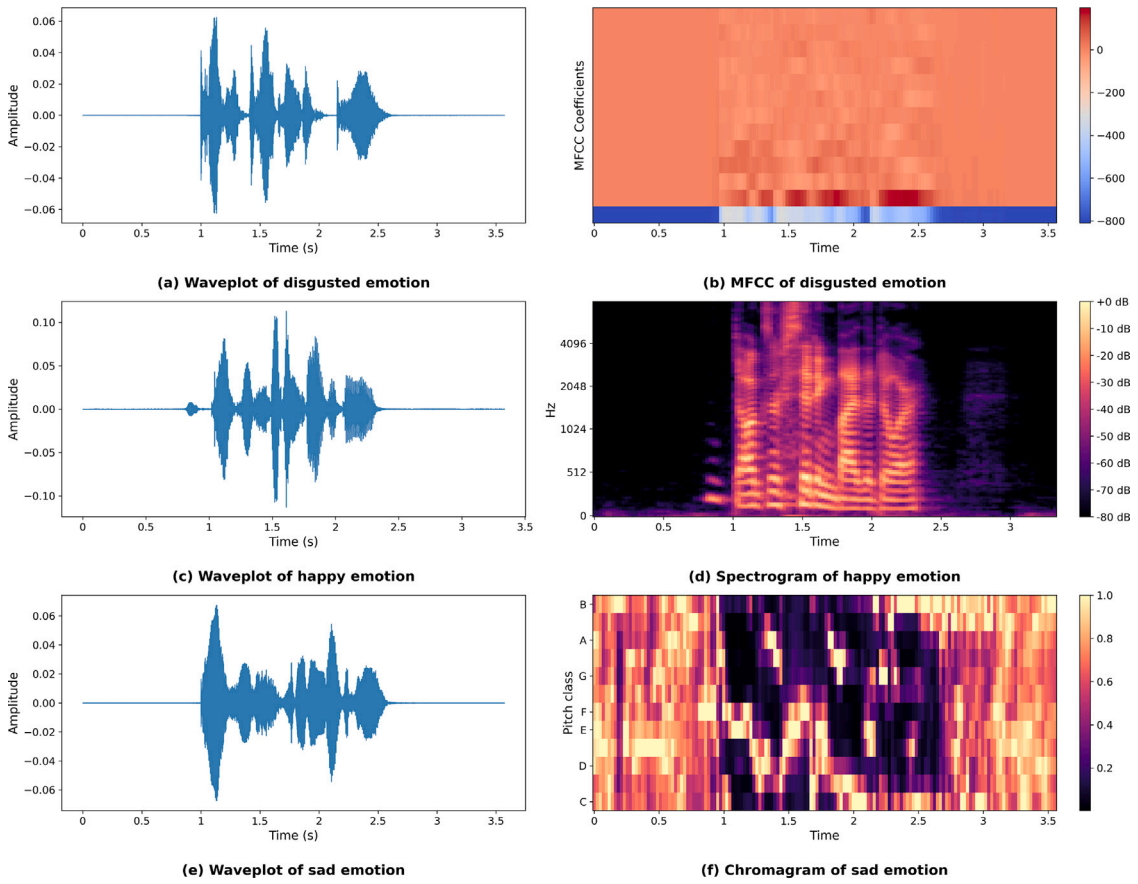


Fig. 3. Visualization of acoustic features for selected emotions. (a) Waveform and (b) MFCC of disgust. (c) Waveform and (d) Spectrogram of happiness. (e) Waveform and (f) Chromagram of sadness. High-arousal emotions exhibit sharp energy bursts, while sadness shows a flatter envelope.

The process of calculating MFCC involves several key steps: Initially, pre-emphasis is applied to enhance the high-frequency components of the speech signal, improving the signal-to-noise ratio, especially when the high-frequency components are weak. The signal is then segmented into short frames, with each frame typically lasting between 20 ms and 40 ms, and a 50% overlap between adjacent frames. Subsequently, the fast Fourier transform (FFT) is applied to convert each frame from the time domain to the frequency domain (Khare et al., 2024), thereby extracting spectral information. This transformation is accomplished by multiplying each sample of the signal by a corresponding complex exponential factor and summing over all samples to obtain the amplitude of the frequency components. Specifically, for a discrete signal $x[m]$ of length M, the FFT process is defined by the following Eq. (17).

$$X[\kappa] = \sum_{m=0}^{M-1} w[m] \cdot x[m] \cdot e^{-j\frac{2\pi}{M}\kappa m}, \quad \kappa = 0, 1, 2, \ldots, M-1, \tag{17}$$

where $x[m]$ is the time-domain signal, $X[\kappa]$ is the frequency-domain signal, and M is the length of the signal (El Ayadi et al., 2011). To avoid spectral leakage, every frame is windowed using a window function. Common window functions include rectangular windows, Hamming windows, and Hanning windows (Wagner et al., 2023). The Hamming window $w[m]$ is defined as in Eq. (18):

$$w[m] = 0.54 - 0.46\cos\left(\frac{2\pi m}{M-1}\right), \quad m = 0, 1, 2, \ldots, M-1. \tag{18}$$

Once each frame is processed with a windowing function, the FFT is performed to extract the frequency spectrum. The mel filter bank is subsequently employed to mimic the human ear's frequency perception characteristics, transforming the speech signal's frequency spectrum into the mel frequency scale, as defined by Eq. (19):

$$f_{Mel} = 2590\log_{10}\left(1 + \frac{f}{700}\right), \tag{19}$$

here, f is the physical frequency and f_{Mel} represents the mel frequency scale.

4.2.2. Chroma features

Chroma features include chroma vector and chromagram, two concepts that are commonly used as feature representations in audio analysis. The chrominance vector consists of 12 elements, each representing the energy of 12 pitches in a specific time period, say a frame. The values of these elements are calculated by accumulating the energy of pitches of different octaves. Chrominance features are highly robust to timbre variations and can effectively extract and represent harmonic and melodic structures in music. It is unique in that notes that differ in pitch by an octave are perceived by the human ear to be similar in timbre. Therefore, pitch can be understood and analyzed in two dimensions: pitch itself and chrominance. In this study, we extracted 12 chrominance features from each audio file for further analysis.

4.2.3. Log-mel spectrogram (LMS)

The log-mel spectrogram is a widely used audio feature extraction method in speech recognition and voiceprint recognition. In this method, the audio signal is first preprocessed and Fourier transformed, and then the mel filter bank is used to extract the energy in different frequency bands. Then, the feature map that can effectively reflect the audio frequency distribution is generated by logarithmic transformation and normalization. Compared with traditional feature extraction methods, log-mel spectrogram shows higher robustness and accuracy in speech recognition tasks. In this study, we extracted 128-dimensional LMS features from each audio file for subsequent analysis.

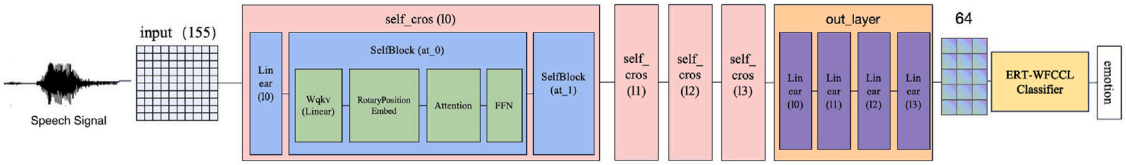


Fig. 4. The module structure of self-attention. The 155-dimensional input features pass through four stacked self-attention blocks and are compressed to a 64-dimensional representation for the ERT-WFCCL classifier.

4.2.4. Zero crossing rate

Zero crossing rate (ZCR), a frequently used feature in audio signal processing, represents the count of instances where a signal crosses the zero value within a specific time interval. Specifically, ZCR quantifies how often an audio signal transitions between positive and negative polarities, providing insight into the signal's spectral properties. ZCR is widely used in many tasks, such as speech and silence discrimination, pitch variation and timbre analysis. In the task of SER, ZCR can effectively capture the frequency fluctuations of the speech signal and is particularly significant in distinguishing various emotions. Emotional changes directly affect the speaker's pronunciation characteristics. For instance, high-arousal emotions, like frustration and enthusiasm, often exhibit swift variations in frequency and high energy, whereas low-arousal emotions, such as calmness and sadness, are marked by slow and steady frequency fluctuations. Integrated with additional attributes, like pitch and intensity, ZCR can provide richer input information for emotion recognition models, thereby enhancing the precision of emotion detection.

4.2.5. Root mean square value

RMS is a commonly used feature in signal processing that represents the mean power or magnitude of a signal. It reflects the signal's overall power by computing the root of the mean squared amplitude over a given time interval. It is calculated as follows:

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2}, \tag{20}$$

where N is the total number of samples in the signal, and x_n denotes the magnitude of the n sampling point. The RMS of the signal was obtained by squaring the amplitude of each sampling point and calculating the mean value after taking the square root. In speech signal processing, RMS is often used to evaluate the energy of audio signals, especially for measuring the loudness and dynamic range of speech. In emotion recognition, RMS reveals the energy differences of speech signals under different emotional expressions. For instance, emotions with intense energy like anger and enthusiasm are typically associated with elevated RMS levels, while low-energy emotions such as tranquility and melancholy, tend to exhibit reduced RMS levels. Therefore, RMS, as an energy feature, can effectively improve the accuracy of speech emotion recognition. The total number of features extracted in this study is 13 MFCCs, 12 chromagrams, 128 LMS, and two ZCR and RMS, forming an initial feature vector of dimension 155 (128 + 13 + 12 + 1 + 1 = 155).

Fig. 3 displays random waveforms from the datasets. Associated spectrogram, MFCC, and chromagram features are also provided for different emotional states. The waveforms of disgust and happiness show sharp energy bursts. Sadness presents a flatter and wider energy envelope. These changes over time are mapped across frequency bands by the spectral features. The visual patterns reveal an uneven distribution of emotional cues across the speech signal. Important acoustic information is often concentrated in short frames. Silence or background noise fills the remaining parts. A direct combination of raw features fails to capture these dynamic changes. The proposed self-attention mechanism assigns higher weights to the most expressive frames. Redundant segments are filtered out by this module before the refined features enter the cognitive classifier.

4.2.6. Self-attention

A feature refinement module based on self-attention is integrated into the model. This design effectively captures inherent temporal dependencies over long distances in speech signals and extracts deep emotional cues. Through a network structure composed of multiple layers, low-level descriptors are transformed into abstract feature representations with higher discriminative power. Given the variable length of speech signals and the limitations of traditional absolute position coding, the module incorporates the rotary position embedding (RoPE) strategy. Standard approaches often rely on stacking absolute position vectors directly. Addressing this limitation, RoPE encodes position data into a block diagonal rotation matrix, which then acts on both the query and key vectors. This ensures that the calculation of attention scores depends entirely on the relative distance between tokens. By building explicit relative position awareness into the system, temporal dynamics between speech frames are captured with greater precision.

The main structure of the model consists of four stacked self-attention blocks. Each block contains a multi-head attention mechanism and a feed-forward neural network, and uses residual connections to promote gradient propagation (Vaswani et al., 2017). The input 155-dimensional acoustic features are first mapped to the high-dimensional space by linear projection, and then the features are reconstructed by four self-attention layers in turn. In this process, the feature dimension is gradually extended to 512 dimensions. This progressive dimension expansion strategy helps the model decouple complex emotional features in higher dimensional semantic space. After deep feature interaction and fusion, in order to adapt to the subsequent weighted fuzzy concept cognitive learning classifier and reduce the computational complexity, the high-dimensional features are compressed into a compact 64-dimensional representation in the output layer. Fig. 4 visually shows the complete architecture of the module with the data flow.

4.3. Model classification

The final stage maps the extracted deep features into the fuzzy concept space for classification using the proposed ERT-WFCCL model. Before the features are passed to the concept-cognitive layer, each conditional attribute is rescaled to the unit interval so that membership values across attributes are directly comparable. The rescaling adopted in this work is min–max normalization, defined as follows:

$$\beta(\mu, \nu) = \frac{\rho(\mu, \nu) - \rho_{\min}(\nu)}{\rho_{\max}(\nu) - \rho_{\min}(\nu)}, \tag{21}$$

where $\beta(\mu, \nu)$ indicates the value of object μ corresponding to attribute ν, and $\rho_{\max}(\nu)$ as well as $\rho_{\min}(\nu)$ denote the maximum and minimum values of ν, respectively. Eq. (21) is therefore a normalization step. Consequently, the fuzzy aspect of the present framework should be understood as structural. The graded behavior of the classifier comes from the hierarchical concept space, the similarity threshold ε, and the progressive concept aggregation in Definition 3.

The normalized data are then fed into a weighted fuzzy concept cognitive learning classifier. The extremely random tree algorithm is used to calculate the feature importance to guide the construction of weighted fuzzy concept lattice. A key hyperparameter in this process is the similarity threshold ϵ, which controls the granularity of the concept space. In this study, we dynamically optimize ϵ in the range of $[0.1, 0.9]$ to balance the generalization ability and discriminative accuracy of the model.

4.4. Implementation details

To ensure the reproducibility of our work and provide a comprehensive understanding of the experimental process, this section details the hardware and software configurations, as well as the specific training strategies and hyperparameter settings employed in this study.

4.4.1. Experimental setup

All experiments were completed on a computer equipped with an Apple M2 processor and 8 GB of memory running macOS Ventura 13.4 operating system. We build deep learning and CCL models based on Python 3.11.7 language and PyTorch framework, and use Librosa library to extract audio features. We fixed the random seed to ensure the reproducibility of the experimental results. The global random seed value is set to 42 for all backends in Python and the NumPy library and PyTorch framework.

4.4.2. Training strategy

In this study, stratified 10-fold cross-validation was adopted to ensure the fairness of model comparison and the robustness of evaluation results. By strictly keeping the sample proportion of each class consistent with the original data set, this strategy effectively eliminates the possible bias caused by random division in imbalanced data sets.

The training of the self-attention network is based on the Adam optimizer, and the parameters are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\hat{\epsilon} = 10^{-8}$. For the optimization of the convergence process, we introduce the Cosine Annealing Learning Rate Schedule strategy, which dynamically adjusts the learning rate η_t of the current round t according to the following equation:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right), \tag{22}$$

where $\eta_{max} = 10^{-5}$ and $\eta_{min} = 10^{-7}$. We set the batch size to 32 and the maximum number of training rounds to 10,000 with an early stopping mechanism.

In view of the class imbalance in EMO-DB and SAVEE datasets, the standard cross-entropy Loss is replaced by Focal Loss, which is defined as follows:

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma}\log(p_t), \tag{23}$$

where p_t represents the predicted probability of the true class. By setting the focus parameter to $\gamma = 2.0$ and the balance parameter to $\alpha = 0.25$, the model is forced to pay more attention to samples that are difficult to classify. In addition, we impose an L1 regularization on the model weights w to promote model sparsity and suppress overfitting according to the following equation:

$$L_{reg} = \lambda \sum_i |w_i|. \tag{24}$$

The regularization factor λ is set to 0.001.

The self-attention front-end uses 8 heads per layer, a hidden dimension of 512, and a dropout rate of 0.1, with residual connections and layer normalization between blocks. Early stopping is triggered when the validation loss does not decrease for 50 consecutive epochs. The ERT classifier is implemented with the scikit-learn defaults: 100 trees, no maximum depth, a minimum of 2 samples to split an internal node, a minimum of 1 sample at a leaf, and the Gini criterion for split quality. Stratified 10-fold splits are generated once with the fixed seed and reused across all classifiers compared in Section 5. Within each iteration, the training folds are further split into a model fitting set for the ERT classifier and a validation set for the SA frontend early stopping, strictly ensuring that the test fold remains completely unseen during all training and feature refinement phases.

5. Results and discussion

In the speech emotion classification task, performance evaluation centers on metrics derived from the confusion matrix. Following the refinement of LLD based acoustic features through the SA mechanism, this section presents a granular analysis of the experimental outcomes. We benchmark the proposed ERT-WFCCL against a comprehensive suite of models, including classical neural networks (CNN, RNN, DNN) and mainstream machine learning methods (SVM, LR, KNN, NB, RF, LDA, DT). All baselines in this section share the SA-refined feature space with ERT-WFCCL. This protocol is designed to compare classifiers on a common feature representation; it is not an end-to-end comparison of deep models in their native operating regime, since deep classifiers such as CNN and RNN are typically designed to ingest raw acoustic input together with their own representation learning stages. The reported numbers should therefore be interpreted as the relative behavior of different classifier heads on shared SA features, not as the upper-bound performance of those deep architectures on these benchmarks. End-to-end comparisons with self-supervised speech models such as wav2vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021) are an important complementary evaluation that we do not pursue in the present work.

Two further structural facts should be kept in mind throughout this section. First, the LMS feature group dominates the input on EMO-DB. LMS alone reaches 77.45% accuracy and accounts for 72.6% of the ERT importance mass, while the full pipeline reaches 81.11%. Removing LMS drops the accuracy by 11.3 percentage points. The framework therefore operates by refining an already strong spectral representation. Second, the SA frontend accounts for the larger share of the improvement over a softmax baseline by 2.55 percentage points, whereas the ERT-WFCCL classifier on top of SA features adds a smaller incremental margin of 0.38 percentage points. These observations frame the rest of this section, as ERT-WFCCL is most usefully read as a lightweight classifier head with a traceable decision path.

5.1. Evaluation metrics

In our research, we assess the effectiveness of various algorithms using key evaluation metrics, including Accuracy, Precision, Recall, and Macro-F1. They are particularly important in the presence of class imbalance, as they offer a thorough evaluation of model performance. For each emotional label L_i, the evaluation depends on four terms: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Each metric is calculated based on the counts of these terms for each class label L_i.

Accuracy represents the overall efficiency of the SER classifier, measuring the fraction of examples that the model correctly predicted. It is calculated as follows

$$\text{Accuracy} = \frac{\sum_i(\text{TP}_i + \text{TN}_i)}{\sum_i(\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i)}. \tag{25}$$

Precision is the fraction of examples predicted to be positive that are actually positive. It is calculated as follows

$$\text{Precision} = \frac{\sum_i \text{TP}_i}{\sum_i(\text{TP}_i + \text{FP}_i)}. \tag{26}$$

Recall represents the fraction of instances that were actually positive and were correctly identified as positive. It is calculated as follows

$$\text{Recall} = \frac{\sum_i \text{TP}_i}{\sum_i(\text{TP}_i + \text{FN}_i)}. \tag{27}$$

Macro-F1 is the harmonic mean of precision and recall, which combines the two into a single comprehensive metric, especially for class imbalance problems. In multi-class emotion recognition tasks, F1 score can capture the characteristics of precision and recall at the same time. It is calculated as follows

$$\text{F1-score}_{macro} = 2 \cdot \frac{\sum_i \text{Precision}_i \cdot \sum_i \text{Recall}_i}{\sum_i \text{Precision}_i + \sum_i \text{Recall}_i}. \tag{28}$$

Table 6
Performance of different models on EMO-DB, RAVDESS, SAVEE and CASIA datasets (Accuracy, Precision, Recall, F1).

Model	Dataset	Accuracy	Precision	Recall	Macro-F1
ERT-WFCCL (Ours)	EMO-DB	**81.11**	**81.12**	**81.12**	**80.99**
	RAVDESS	**66.32**	**66.16**	**66.32**	65.36
	SAVEE	**74.58**	**76.09**	**74.58**	**74.05**
	CASIA	76.58	76.51	76.58	76.49
CNN	EMO-DB	77.75	77.90	77.76	77.50
	RAVDESS	64.93	64.74	64.93	64.55
	SAVEE	74.37	75.18	74.37	73.98
	CASIA	71.75	71.87	71.75	71.33
RNN	EMO-DB	68.43	68.33	68.41	67.83
	RAVDESS	65.06	62.39	65.06	63.29
	SAVEE	69.37	70.30	69.37	68.27
	CASIA	43.50	44.85	43.50	43.22
DNN	EMO-DB	67.87	67.95	67.89	67.32
	RAVDESS	64.51	62.24	64.51	62.13
	SAVEE	66.04	67.12	66.04	63.29
	CASIA	75.41	75.73	75.41	75.37
SVM	EMO-DB	74.18	74.26	74.21	73.47
	RAVDESS	57.43	61.79	57.43	59.16
	SAVEE	71.04	72.30	71.04	69.07
	CASIA	**77.83**	**77.86**	**77.83**	**77.78**
LR	EMO-DB	72.89	72.51	72.90	72.37
	RAVDESS	66.11	65.98	66.11	**65.78**
	SAVEE	72.50	72.48	72.50	71.90
	CASIA	77.41	77.63	77.41	77.50
KNN	EMO-DB	68.58	68.53	68.60	68.48
	RAVDESS	59.86	60.16	59.86	59.30
	SAVEE	68.12	68.11	68.12	67.39
	CASIA	68.66	69.82	68.66	68.05
NB	EMO-DB	47.09	53.56	47.10	46.34
	RAVDESS	39.86	43.91	39.86	38.53
	SAVEE	49.79	51.59	49.79	47.39
	CASIA	47.16	51.20	47.16	46.09
RF	EMO-DB	69.70	68.84	69.72	68.34
	RAVDESS	61.04	61.84	61.04	60.33
	SAVEE	72.70	73.87	72.70	71.90
	CASIA	72.66	72.24	72.66	72.38
LDA	EMO-DB	71.42	71.19	71.40	71.24
	RAVDESS	60.00	61.39	60.00	59.92
	SAVEE	49.37	46.91	49.37	46.82
	CASIA	75.16	76.05	75.16	75.35
DT	EMO-DB	54.74	54.05	54.77	54.33
	RAVDESS	40.83	40.99	40.83	40.86
	SAVEE	55.41	56.84	55.41	55.40
	CASIA	56.33	56.33	56.33	56.30

5.2. Comparative analysis of classification performance

The accuracy, precision, recall, and F1 score of each method across the four datasets are summarized in Table 6, where all classifiers are trained on the same self-attention features. ERT-WFCCL achieves the best accuracy on EMO-DB and SAVEE, and yields 66.32% on RAVDESS, close to the 66.11% obtained by LR. On CASIA, its score of 76.58% trails SVM and LR by roughly one percentage point. These results suggest that the fuzzy concept space is particularly useful when emotion categories overlap in the feature space. For EMO-DB and SAVEE, where several emotions share similar arousal levels, fuzzy membership offers a softer decision boundary than hyperplane-based classifiers. CASIA samples, by contrast, are acoustically cleaner and can be well separated by linear models, so the concept layer brings little additional benefit on this dataset. The confusion matrices on EMO-DB are presented in Fig. 5. ERT-WFCCL attains high true positive rates for Anger and Sadness, while most of the remaining errors occur between Happiness and Anger, two high-arousal emotions with similar spectral envelopes.

Table 7 breaks down the EMO-DB performance by emotion class. Focal loss is used during feature extraction to give more weight to minority samples. Despite containing only 46 instances, the Disgust class still attains an F1 score of 75%, while majority classes such as Anger and Sadness reach 89% and 88%, respectively. The per-class F1 scores fall within a relatively narrow range of 72% to 89%, indicating that the model does not collapse onto majority classes. A similar behavior is also observed on SAVEE, where the minority Neutral class does not dominate the predictions.

Table 7
Class-wise performance metrics of the proposed ERT-WFCCL model on the EMO-DB dataset.

Emotion	Precision (%)	Recall (%)	F1-score (%)	Support
Anger	87	91	89	127
Boredom	76	77	76	81
Disgust	76	74	75	46
Fear	75	84	79	69
Happiness	80	66	72	71
Sadness	89	87	88	62
Neutral	81	80	80	79
Accuracy			**81.11**	535
Macro Avg	80	80	80	535
Weighted Avg	81	81	81	535

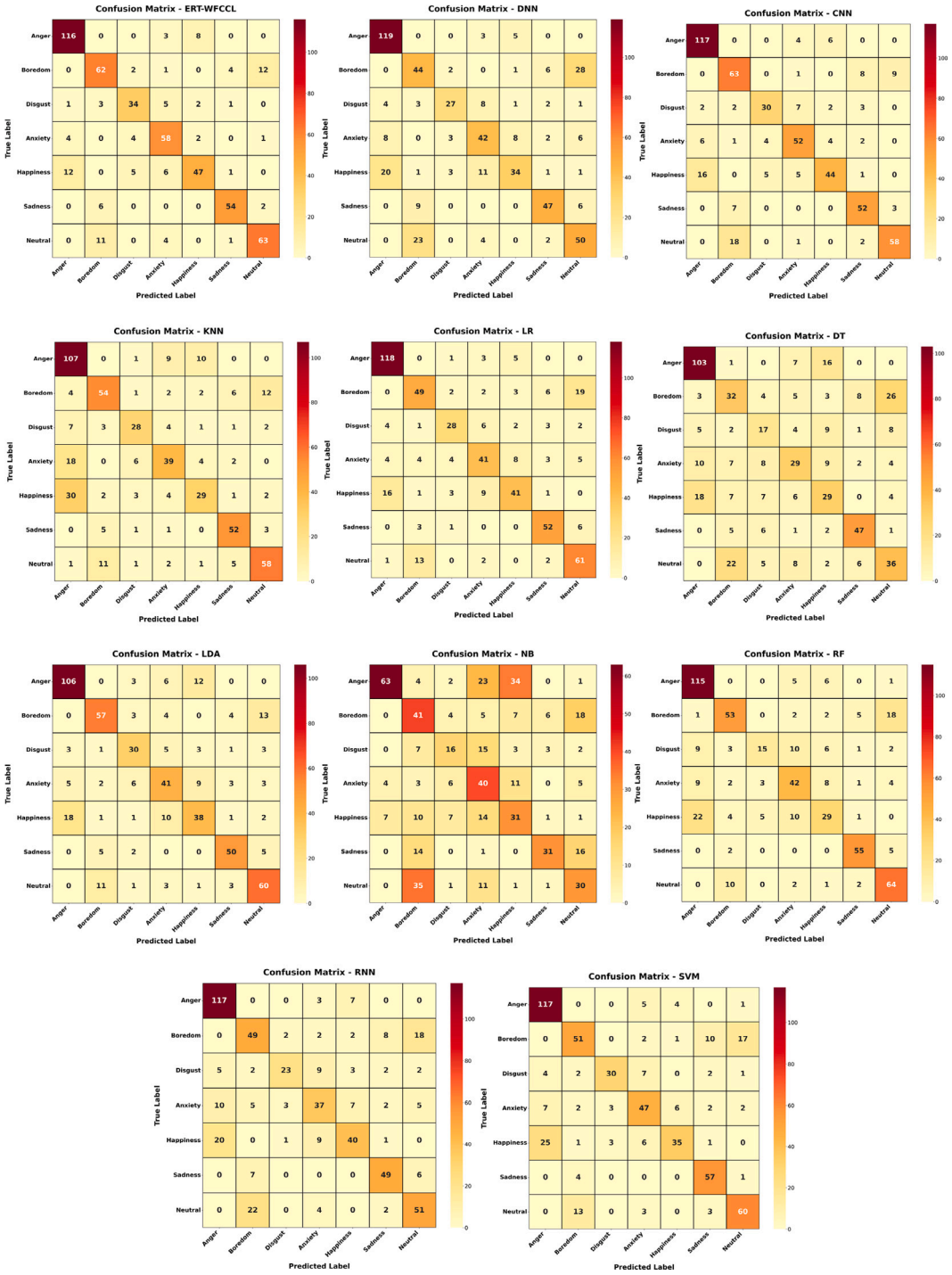


Fig. 5. Confusion matrix of all the compared experimental models in the EMO-DB dataset.

The deep baselines exhibit larger performance swings across datasets. CNN reaches 77.75% on EMO-DB and 74.37% on SAVEE, yet drops to 64.93% on RAVDESS, while RNN falls to 43.50% on CASIA. Such instability is commonly observed when deep classifiers are trained on small datasets containing only a few hundred to a thousand samples. By comparison, ERT-WFCCL remains within a narrower range of 66%–81% across the four datasets, partly because the fuzzy concept classifier has fewer trainable parameters than the deep baselines and is therefore less prone to overfitting on small folds. However, that the stability reported here is measured within curated benchmarks and may not extend to naturalistic recording conditions.

Linear classifiers perform strongly on CASIA, with SVM and LR reaching 77.83% and 77.41% respectively, both slightly above ERT-WFCCL at 76.58%. The gap is on the order of one percentage point, and the paired t-test in Section 5.4 returns $p = 0.34$ for SVM and $p = 0.54$ for LR, indicating that the differences are not statistically significant. A plausible explanation is that CASIA samples are acoustically cleaner than those of the other three datasets, a condition that favors linear decision boundaries. The pattern is reversed on RAVDESS,

Table 8
Comparison of model performance under stratified 10-fold CV and leave-one-speaker-out (LOSO) CV strategies.

Dataset	Validation strategy	Accuracy	Precision	Recall	Macro-F1	Acc. Drop (Δ)
EMO-DB	10-Fold CV	81.11	81.12	81.12	80.99	↓ 1.30%
	LOSO	79.81	79.71	79.81	79.56	
RAVDESS	10-Fold CV	66.32	66.16	66.32	65.36	↓ 1.67%
	LOSO	64.65	64.65	64.65	64.25	
SAVEE	10-Fold CV	74.58	76.09	74.58	74.05	↓ 3.13%
	LOSO	71.45	72.04	72.04	70.32	
CASIA	10-Fold CV	76.58	76.51	76.58	76.49	↓ 3.92%
	LOSO	72.66	72.24	72.66	72.38	

Table 9
Feature ablation on EMO-DB. Performance of individual LLD feature groups, full combination, and leave-LMS-out configuration. All settings share the same self-attention front-end and ERT-WFCCL classifier.

Input Features (Dim)	Accuracy (%)	F1-score (%)
MFCC (13)	68.37 ± 4.20	64.85
LMS (128)	77.45 ± 6.40	72.99
Chromagram (12)	39.13 ± 5.65	34.14
ZCR + RMS (2)	45.83 ± 6.85	47.06
w/o LMS (27)	69.78 ± 3.60	67.53
ALL Combined (155)	**78.56 ± 5.73**	**75.18**

Table 10
Impact of the self-attention module on EMO-DB.

Model	Accuracy (%)	F1-score (%)
LLDs + SA	80.73 ± 4.75	80.55
LLDs + SA + ERT-WFCCL	**81.11 ± 4.99**	**80.99**
LLDs + ERT-WFCCL	78.56 ± 5.73	75.18

where SVM drops to 57.43% and NB to 39.86%, whereas ERT-WFCCL maintains an accuracy of 66.32%. Linear boundaries appear to be a poor fit when acoustic variability is high; the fuzzy concept space copes better with such conditions, although it remains marginally below the linear baselines on the cleaner CASIA data. The framework also produces a decision trace for each sample. Each test sample is assigned to the nearest progressive weighted fuzzy concept according to the Euclidean distance defined in Eq. (16), and the contribution of each acoustic attribute to that distance is quantified by the ERT importance weights.

5.3. Speaker-independent evaluation

We conducted a leave-one-speaker-out (LOSO) cross-validation experiment across the four datasets to evaluate the generalization capability of the model and verify that the classifier does not overfit to speaker-specific timbre characteristics. The experimental setup for both the SA feature extraction and the ERT-WFCCL classifier was identical to the stratified 10-fold cross-validation discussed previously. The key difference is that the LOSO strategy tests the model exclusively on unseen speakers.

As detailed in Table 8, accuracy decreases by 1.30% on EMO-DB, 1.67% on RAVDESS, 3.13% on SAVEE, and 3.92% on CASIA when moving from stratified 10-fold to LOSO. The drops on SAVEE and CASIA are larger than those on the other two datasets, reflecting greater speaker variation in these corpora. The four benchmarks consist of acted speech recorded in controlled conditions.

5.4. Ablation studies and statistical significance

The contribution of each acoustic feature group is first examined on EMO-DB. Table 9 reports that LMS alone reaches 77.45% and stands out as the strongest single feature group. This is followed by MFCC at 68.37%, while Chromagram and ZCR+RMS lag at 39.13% and 45.83% respectively. Concatenating all groups raises the accuracy to 78.56%, and the full SA and ERT-WFCCL pipeline further pushes it to 81.11%. When LMS is removed from the full pipeline, the accuracy drops to 69.78% and the F1 score to 67.53%. This decrease of 11.3 points confirms LMS as the carrier of the most discriminative signal. The full framework adds approximately 3.7 percentage points on top of using only LMS, increasing from 77.45% to 81.11%. This margin is the upper bound of what the SA frontend and ERT-WFCCL classifier can jointly contribute beyond the dominant feature family. Fig. 6 presents the ERT importance scores which paint a consistent picture. LMS accounts for 72.6% of the total importance while occupying 82.5% of the input dimensions, and ZCR, although having only two dimensions, still ranks among the top 20.

Table 10 separates the contributions of the SA module and the ERT-WFCCL classifier on the EMO-DB dataset. Removing SA and feeding raw LLDs into ERT-WFCCL reduces the accuracy from 81.11% to 78.56%, resulting in a drop of 2.55 percentage points. Replacing ERT-WFCCL with a softmax head on top of the SA features only lowers the accuracy to 80.73%, a drop of 0.38 percentage points. This asymmetry indicates that the SA frontend is the dominant contributor on this dataset, while ERT-WFCCL provides a smaller incremental contribution along with a decision trace for each sample.

A limitation exists in the granularity of this ablation. The current design jointly varies two factors inside the classifier, namely the ERT derived attribute weights and the fuzzy concept structure including progressive concept aggregation, similarity threshold ϵ, and nearest concept matching. The evaluation reports their joint effect against a softmax baseline. The ablation does not isolate the fuzzy concept structure from the ERT weighting itself. Comparing ERT-WFCCL to a softmax classifier on SA features confounds the contribution of the structured concept space with that of the weighted distance metric. A cleaner decomposition would compare the ERT weighted distance to the nearest concept, the ERT weighted distance to class centroids without concept structure, and the unweighted nearest concept matching, all using the same SA features. Such a decomposition would clarify whether the small margin of 0.38 percentage points originates from the concept space or from the weighted distance, which remains a target for future work.

Two-tailed paired t-tests are applied to the fold-wise accuracies of each dataset, with Bonferroni correction at $\alpha = 0.005$ to account for the ten model comparisons per dataset. Table 11 lists the fold-wise accuracies on EMO-DB as an illustration, and the test results across all four datasets are summarized in Table 12.

As shown in Table 12, on EMO-DB ERT-WFCCL outperforms nine of the ten baselines at $\alpha = 0.05$ and eight of them under Bonferroni correction. The single exception is the comparison against CNN. Although the mean accuracy gap of 3.36 percentage points is the largest improvement observed on this dataset against a strong deep baseline, the paired t test returns $p = 0.074$, which does not reach the conventional $\alpha = 0.05$ threshold. The numerical advantage on EMO-DB therefore does not constitute a statistically established advantage over CNN, whereas the advantage against the remaining nine baselines is statistically supported.

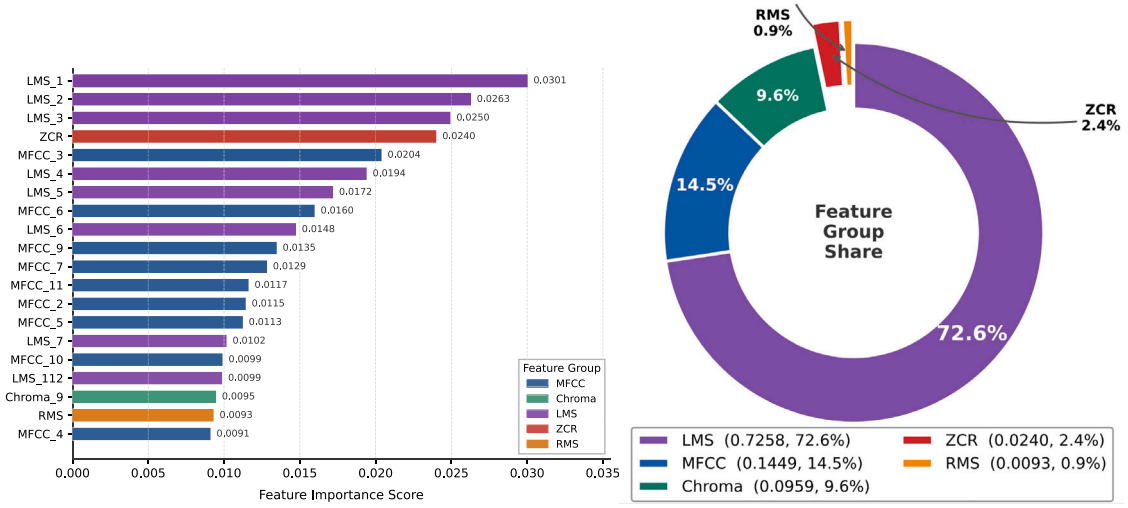


Fig. 6. ERT feature importance analysis on the EMO-DB dataset. Left: top 20 individual feature importance scores. Right: aggregated importance by feature group.

Table 11
The fold-wise accuracy (%) of different models on the EMO-DB dataset.

Model	Mean Acc	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10
ERT-WFCCL	**81.11**	**81.48**	**81.48**	81.48	**83.33**	**83.33**	**77.36**	**90.57**	75.47	**84.91**	71.70
CNN	77.75	79.63	79.63	**83.33**	70.37	79.63	71.70	79.25	**77.36**	83.02	**73.58**
RNN	68.43	64.81	64.81	64.81	61.11	74.07	75.47	73.58	69.81	73.58	62.26
DNN	67.87	61.11	61.11	74.07	62.96	72.22	75.47	66.04	67.92	77.36	60.38
SVM	74.18	75.93	75.93	79.63	70.37	81.48	71.70	75.47	77.36	71.70	62.26
LR	72.89	68.52	70.37	79.63	70.37	77.78	71.70	69.81	73.58	79.25	67.92
LDA	71.42	77.78	64.81	70.37	64.81	68.52	73.58	79.25	75.47	75.47	64.15
KNN	68.58	72.22	72.22	74.07	66.67	68.52	67.92	73.58	69.81	56.60	64.15
RF	69.70	75.93	74.07	74.07	61.11	72.22	67.92	69.81	73.58	69.81	58.49
Decision Tree	54.74	48.15	57.41	55.56	64.81	62.96	52.83	52.83	47.17	52.83	52.83
NB	47.09	40.74	42.59	51.85	46.30	62.96	49.06	32.08	52.83	49.06	43.40

Performance on the other three datasets presents a different trend. On RAVDESS, the accuracy gaps are 0.21 percentage points against LR, 1.25 against RNN, 1.39 against CNN, and 1.80 against DNN. These are all within two percentage points and none is statistically significant. On SAVEE, the gaps are 0.21 percentage points against CNN, 1.88 against RF, 2.08 against LR, 3.54 against SVM, and 5.21 against RNN, which likewise fail to reach significance. On CASIA, ERT-WFCCL trails SVM by 1.25 percentage points and LR by 0.83 percentage points, and these differences are also not significant. Overall, significant gains are concentrated on EMO-DB and against weaker baselines. Against the strongest neural and linear competitors on RAVDESS, SAVEE, and CASIA, ERT-WFCCL is statistically indistinguishable from the leading method and marginally below it on CASIA. The cross dataset behavior indicates that ERT-WFCCL is competitive with the strongest baselines under the present protocol, achieving significantly better results than most competitors primarily on the EMO-DB dataset.

Table 13 reports the single-fold training time and per-sample inference time for all compared methods on the EMO-DB dataset to provide a practical assessment of computational efficiency. Note that all methods share the same SA-based feature extraction front-end, and only the classifier-specific costs are compared here.

Among all methods, ERT-WFCCL takes the longest to train, at 22.54 s per fold, mainly because of the construction and traversal of the fuzzy concept space. Traditional machine learning methods finish training in under one second, and deep learning classifiers typically need 1–3 s. That said, training is a one-time offline expense and does not affect deployment. On the inference side, ERT-WFCCL processes each sample in about 33.56 ms, which is slower than traditional methods but still well below the 100-millisecond threshold generally considered acceptable for real-time human–computer interaction. The per-sample inference cost of 33.56 ms is higher than that of the traditional classifiers, yet still falls below the 100 ms latency commonly cited for real-time interactive systems. This cost reflects the construction and traversal of the fuzzy concept space during prediction.

5.5. Parameter sensitivity analysis

The depth of feature abstraction directly determines the ability of the model to capture emotional information, so the number of SA layers is the primary factor affecting the performance. As illustrated in Fig. 7(a), the experiment compares the influence of the number of layers from 1 to 5 on the accuracy, and the results show that the model performance does not increase linearly with the increase of the number of layers. When the number of layers is 1, the accuracy is only 76.81%, which indicates that the shallow network is difficult to fully extract long-distance dependencies in high-order acoustic features. With the number of layers increasing to 4, the accuracy of the model reaches a peak of 81.11%, which proves that the network with appropriate depth can effectively aggregate key emotional segments. However, when increasing to 5 layers, the performance drops to 79.99%. This phenomenon is usually attributed to the overfitting problem on small sample datasets, and too deep networks introduce unnecessary parameter noise and weaken the generalization ability of the model. In this study, the number of SA layers is locked to four layers to balance feature expression and model complexity.

The similarity threshold ε governs the granularity of the concept space, in the sense that a small ε preserves many fine grained concepts whereas a large ε merges them into coarser units. The corresponding accuracy on EMO-DB across $\varepsilon \in [0.1, 0.9]$ is plotted in Fig. 7b, where the curve peaks at $\varepsilon = 0.3$ with 81.11%, remains above 77% for $\varepsilon \leq 0.5$, and falls to 66.37% at $\varepsilon = 0.9$. It should be noted that the search for ε

Table 12
Two-tailed paired t-test results of ERT-WFCCL versus competing models across all four datasets, with Bonferroni correction at $\alpha = 0.005$ accounting for 10 comparisons per dataset. Within each dataset, models are sorted by absolute mean difference from ERT-WFCCL in ascending order.

Dataset	Model	Mean Diff (%)	t-statistic	p-value	Sig. (α=0.05)	Bonf. (α=0.005)
EMO-DB	CNN	+3.36	2.017	0.0744	No	No
EMO-DB	SVM	+6.93	3.902	0.0036	Yes	Yes
EMO-DB	LR	+8.22	4.284	0.0020	Yes	Yes
EMO-DB	LDA	+9.69	5.084	0.0007	Yes	Yes
EMO-DB	RF	+11.41	5.538	0.0004	Yes	Yes
EMO-DB	KNN	+12.54	5.801	0.0003	Yes	Yes
EMO-DB	RNN	+12.68	6.457	0.0001	Yes	Yes
EMO-DB	DNN	+13.25	5.534	0.0004	Yes	Yes
EMO-DB	DT	+26.37	12.889	<0.0001	Yes	Yes
EMO-DB	NB	+34.02	9.819	<0.0001	Yes	Yes
RAVDESS	LR	+0.21	0.076	0.9410	No	No
RAVDESS	RNN	+1.25	0.423	0.6822	No	No
RAVDESS	CNN	+1.39	0.427	0.6795	No	No
RAVDESS	DNN	+1.80	0.674	0.5175	No	No
RAVDESS	RF	+5.28	2.524	0.0326	Yes	No
RAVDESS	LDA	+6.32	3.426	0.0076	Yes	No
RAVDESS	KNN	+6.46	2.565	0.0304	Yes	No
RAVDESS	SVM	+8.89	4.089	0.0027	Yes	Yes
RAVDESS	DT	+25.49	12.435	<0.0001	Yes	Yes
RAVDESS	NB	+26.46	10.398	<0.0001	Yes	Yes
SAVEE	CNN	+0.21	0.098	0.9239	No	No
SAVEE	RF	+1.88	0.657	0.5275	No	No
SAVEE	LR	+2.08	0.775	0.4583	No	No
SAVEE	SVM	+3.54	1.012	0.3381	No	No
SAVEE	RNN	+5.21	1.659	0.1316	No	No
SAVEE	KNN	+6.46	2.379	0.0413	Yes	No
SAVEE	DNN	+8.54	3.568	0.0060	Yes	No
SAVEE	DT	+19.17	5.863	0.0002	Yes	Yes
SAVEE	NB	+24.79	12.875	<0.0001	Yes	Yes
SAVEE	LDA	+25.21	9.670	<0.0001	Yes	Yes
CASIA	SVM	−1.25	−1.009	0.3393	No	No
CASIA	LR	−0.83	−0.644	0.5356	No	No
CASIA	DNN	+1.17	0.760	0.4665	No	No
CASIA	LDA	+1.42	0.733	0.4825	No	No
CASIA	RF	+3.92	2.404	0.0397	Yes	No
CASIA	CNN	+4.84	3.211	0.0106	Yes	No
CASIA	KNN	+7.92	4.437	0.0016	Yes	Yes
CASIA	DT	+20.25	12.569	<0.0001	Yes	Yes
CASIA	NB	+29.42	15.992	<0.0001	Yes	Yes
CASIA	RNN	+33.09	10.311	<0.0001	Yes	Yes

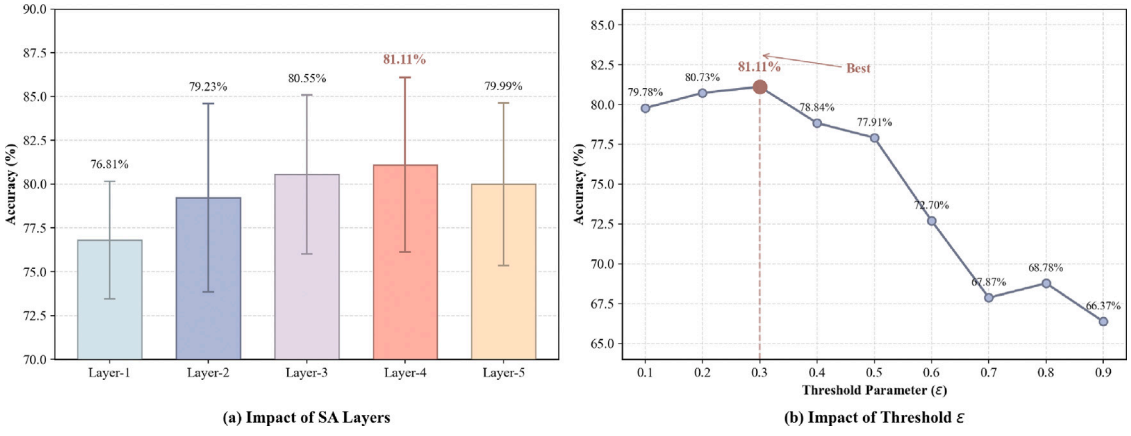


Fig. 7. Parameter sensitivity analysis on the EMO-DB dataset. (a) The impact of stacking different numbers of Self-Attention layers on recognition accuracy. (b) The trend of model performance with varying threshold parameter ϵ in the ERT-WFCCL classifier.

Table 13
Computational cost comparison of different classifiers on the EMO-DB dataset (Apple M2 processor).

Model	Training time (s)	Inference time (ms/sample)
ERT-WFCCL (Ours)	22.54	33.56
CNN	3.19	23.58
DNN	1.26	22.50
RNN	1.21	23.17
RF	0.19	2.11
LR	0.09	0.03
LDA	0.03	0.03
DT	0.03	0.03
SVM	0.02	0.09
KNN	<0.01	0.31
NB	<0.01	0.13

is performed on the same EMO-DB folds used to report the main accuracy. Because nested cross validation is not implemented, the EMO-DB number reflects the selection of ϵ on the evaluation data. Additionally, the value $\epsilon = 0.3$ obtained from the EMO-DB search is transferred to RAVDESS, SAVEE, and CASIA without retuning. Because per dataset sensitivity curves are not provided for the other three benchmarks, the generalization of $\epsilon = 0.3$ across corpora is not empirically verified in this work. The accuracies reported on the remaining datasets therefore represent performance under a single EMO-DB tuned threshold.

6. Conclusions and future work

This paper studies a pipeline level integration of concept cognitive learning for speech emotion recognition. The proposed ERT-WFCCL framework utilizes a self attention frontend to refine acoustic LLDs, followed by a weighted fuzzy CCL classifier with ERT derived attribute weights that matches each utterance to its closest progressive concept. The approach is evaluated on four public benchmarks spanning English, German, and Chinese.

Experimental results demonstrate that ERT-WFCCL achieves the highest accuracy among eleven classifiers on EMO-DB, significantly outperforming most baselines. On RAVDESS, SAVEE, and CASIA, the framework remains competitive with strong neural and linear baselines, although the performance differences do not reach statistical significance. Ablation studies confirm that the self attention frontend drives the majority of the improvement over raw spectral representations. The cognitive classifier provides a smaller incremental accuracy gain while producing a transparent decision trace for each sample.

Several technical limitations present opportunities for future research. The current ablation does not isolate the fuzzy concept space from the ERT weighting mechanism, and the min max normalization serves a purely structural role. Additionally, hyperparameters including the similarity threshold and the progressive weighting scheme are not exhaustively cross validated across all corpora or compared against all potential alternatives. Finally, the evaluation protocol measures the relative performance of classifier heads on shared refined features rather than comparing end to end architectures in their native configurations.

CRediT authorship contribution statement

Weihua Xu: Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Kaiping Hu:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62376229 and in part by the Natural Science Foundation of Chongqing, China under Grant CSTB2023NSCQ-LZX0027.

Data availability

The link of all used data for the research described have been shared in the article.

References

Ahmed, M.R., Islam, S., Islam, A.M., Shatabda, S., 2023. An ensemble 1D-CNN-LSTM-GRU model with data augmentation for speech emotion recognition. Expert Syst. Appl. 218, 119633.

Akçay, M.B., Oğuz, K., 2020. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. Speech Commun. 116, 56–76.

Baevski, A., Zhou, Y., Mohamed, A., Auli, M., 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. 33, 12449–12460.

Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W.F., Weiss, B., 2005. A database of German emotional speech. In: Proceedings of Interspeech 2005. ISCA, Lisbon, Portugal, pp. 1517–1520.

Cummins, N., Scherer, S., Krajewski, J., Schnieder, S., Epps, J., Quatieri, T.F., 2015. A review of depression and suicide risk assessment using speech analysis. Speech Commun. 71, 10–49.

de Lope, J., Graña, M., 2023. An ongoing review of speech emotion recognition. Neurocomputing 528, 1–11.

El Ayadi, M., Kamel, M.S., Karray, F., 2011. Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognit. 44 (3), 572–587.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63, 3–42.

Guo, D., Xu, W., 2023. Fuzzy-based concept-cognitive learning: An investigation of novel approach to tumor diagnosis analysis. Inform. Sci. 639, 118998.

Guo, D., Xu, W., Ding, W., Yao, Y., Wang, X., Pedrycz, W., Qian, Y., 2024. Concept-cognitive learning survey: Mining and fusing knowledge from data. Inf. Fusion 109, 102426.

Guo, D., Xu, W., Qian, Y., Ding, W., 2023a. Fuzzy-granular concept-cognitive learning via three-way decision: performance evaluation on dynamic knowledge discovery. IEEE Trans. Fuzzy Syst. 32 (3), 1409–1423.

Guo, D., Xu, W., Qian, Y., Ding, W., 2023b. M-FCCL: Memory-based concept-cognitive learning for dynamic fuzzy data classification and knowledge fusion. Inf. Fusion 100, 101962.

Hashem, A., Arif, M., Alghamdi, M., 2023. Speech emotion recognition approaches: A systematic review. Speech Commun. 154, 102974.

Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A., 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 3451–3460.

Jackson, P., Haq, S., 2014. Surrey audio-visual expressed emotion (SAVEE) database. URL: http://personal.ee.surrey.ac.uk/Personal/P.Jackson/SAVEE/. University of Surrey, Guildford, UK.

Joysingh, S.J., Vijayalakshmi, P., Nagarajan, T., 2025. Significance of chirp MFCC as a feature in speech and audio applications. Comput. Speech Lang. 89, 101713.
Khare, S.K., Blanes-Vidal, V., Nadimi, E.S., Acharya, U.R., 2024. Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations. Inf. Fusion 102, 102019.
Latif, S., Rana, R., Khalifa, S., Jurdak, R., Qadir, J., Schuller, B., 2021. Survey of deep representation learning for speech emotion recognition. IEEE Trans. Affect. Comput. 14 (2), 1634–1654.
Li, J., Huang, C., Qi, J., Qian, Y., Liu, W., 2017. Three-way cognitive concept learning via multi-granularity. Inform. Sci. 378, 244–263.
Li, D., Liu, J., Yang, Z., Sun, L., Wang, Z., 2021. Speech emotion recognition using recurrent neural networks with directional self-attention. Expert Syst. Appl. 173, 114683.
Li, Y., Tao, J., Chao, L., Bao, W., Liu, Y., 2017. CHEAVD: a Chinese natural emotional audio–visual database. J. Ambient. Intell. Humaniz. Comput. 8, 913–924.
Liu, M., Raj, A.N.J., Rajangam, V., Ma, K., Zhuang, Z., Zhuang, S., 2024. Multiscale-multichannel feature extraction and classification through one-dimensional convolutional neural network for speech emotion recognition. Speech Commun. 156, 103010.
Livingstone, S.R., Russo, F.A., 2018. The Ryerson Audio-Visual Database of Emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS One 13 (5), e0196391.
Luengo, I., Navas, E., Hernáez, I., Sánchez, J., 2005. Automatic emotion recognition using prosodic parameters. In: Interspeech. ISCA, Lisbon, Portugal, pp. 493–496.
Mi, Y., Liu, W., Shi, Y., Li, J., 2022. Semi-supervised concept learning by concept-cognitive learning and concept space. IEEE Trans. Knowl. Data Eng. 34 (5), 2429–2442.
Mi, Y., Shi, Y., Li, J., Liu, W., Yan, M., 2020. Fuzzy-based concept learning method: Exploiting data with fuzzy conceptual clustering. IEEE Trans. Cybern. 52 (1), 582–593.
Nwe, T.L., Foo, S.W., De Silva, L.C., 2003. Speech emotion recognition using hidden Markov models. Speech Commun. 41 (4), 603–623.
Singh, P.K., Cherukuri, A.K., Li, J., 2017. Concepts reduction in formal concept analysis with fuzzy setting using Shannon entropy. Int. J. Mach. Learn. Cybern. 8, 179–189.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.
Wagner, J., Triantafyllopoulos, A., Wierstorf, H., Schmitt, M., Burkhardt, F., Eyben, F., Schuller, B.W., 2023. Dawn of the transformer era in speech emotion recognition: closing the valence gap. IEEE Trans. Pattern Anal. Mach. Intell. 45 (9), 10745–10759.
Xu, W., Guo, D., Mi, J., Qian, Y., Zheng, K., Ding, W., 2023. Two-way concept-cognitive learning via concept movement viewpoint. IEEE Trans. Neural Netw. Learn. Syst. 34 (10), 6798–6812.
Xu, W., Zhang, C., 2025. RF-WFCCL: A random forest-driven weighted fuzzy concept cognitive learning. Fuzzy Sets and Systems 109724.
Yahia, S.B., Arour, K., Slimani, A., Jaoua, A., 2000. Discovery of compact rules in relational databases. Inf. Sci. J. 4 (3), 497–511.
Zhang, S., Guo, P., Zhang, J., Wang, X., Pedrycz, W., 2012. A completeness analysis of frequent weighted concept lattices and their algebraic properties. Data Knowl. Eng. 81, 104–117.
Zhang, C., Tsang, E.C., Xu, W., Lin, Y., Yang, L., 2023. Incremental concept-cognitive learning approach for concept classification oriented to weighted fuzzy concepts. Knowl.-Based Syst. 260, 110093.
Zhang, S., Yang, Y., Chen, C., Zhang, X., Leng, Q., Zhao, X., 2024. Deep learning-based multimodal emotion recognition from audio, visual, and text modalities: A systematic review of recent advancements and future prospects. Expert Syst. Appl. 237, 121692.
Zhao, J., Mao, X., Chen, L., 2019. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. Biomed. Signal Process. Control. 47, 312–323.