Emotion recognition in conversation using SDT

Riccardo Berni
r.berni@studenti.unipisa.it

Michela Faella
m.faella1@studenti.unipisa.it

Jessica Ferrari
j.ferrari1@studenti.unipisa.it

Alessandro Guerriero
a.guerriero3@studenti.unipisa.it

Margherita Merialdo
m.merialdo@studenti.unipisa.it

Abstract

We present a Transformer-based model for multimodal Emotion Recognition in Conversation, building on the Self-Distillation Transformer (SDT). Our approach uses text and audio modalities, intra/inter-modal attention, gated fusion, and a self-distillation mechanism that enhances unimodal learning without a separate teacher model. We further explore an extension, BiosERC, that integrates speaker personality traits generated by large language models. Experiments on the IEMOCAP dataset show that our SDT variant does not match the performance of the original model—a result that was expected given our different architectural configurations and significantly lower computational resources.

1 Introduction

Emotion Recognition in Conversations (ERC) is a key task in affective computing, with implications for AI empathy and human-computer interaction. It involves identifying emotions expressed across dialogues, considering both content and context. While early work relied on unimodal data, recent approaches emphasize multimodal learning to better capture emotional signals [10]. However, challenges persist in modality fusion and modeling speaker dynamics.

In this project, we reimplement and analyze the Self-Distillation Transformer (SDT) model [8], a state-of-the-art framework leveraging intra-/inter-modal Transformers, hierarchical fusion, and self-distillation to improve unimodal learning from multimodal data. Our adaptation focuses on text and audio modalities, omitting visual inputs for efficiency.

Our main contributions are: *(i)* a near-faithful reimplementation of SDT for text and audio, *(ii)* an analysis of self-distillation effects on unimodal performance, *(iii)* a modular preprocessing and evaluation pipeline for the IEMOCAP dataset, and *(iv)* integration of speaker personality embeddings to assess their impact.

The rest of the report is organized as follows: Section 2 reviews ERC and the SDT framework; Section 3 details our implementation; Section 4 describes the experimental setup; Section 5 discusses results; and Section 6 concludes with insights and future directions.

2 Background

Emotion Recognition in Conversations (ERC) is an emerging field within affective computing, with applications ranging from empathetic dialogue systems to mental health support. Traditional ERC

HLT final project report (2025).

research has focused primarily on textual modalities, often neglecting the rich information conveyed through visual and acoustic cues. Recent work [11] [5] emphasizes the importance of multimodal fusion to capture inter- and intra-modal interactions and to model speaker- and context-sensitive dependencies more effectively.

Several foundational models have been instrumental in advancing ERC. Conversational Memory Network (CMN) [3] employs Gated Recurrent Units (GRUs) to model the conversational context of each speaker in dyadic interactions. It constructs separate memory networks for each speaker, enabling the capture of speaker-specific contextual information. An attention mechanism is then utilized to integrate these memories, facilitating the modeling of inter-speaker dependencies. Interactive Conversational Memory Network (ICON) [3], building upon CMN, incorporates an additional GRU to explicitly model inter-speaker emotional influences. This enhancement allows the model to better capture the dynamic emotional interplay between speakers throughout a conversation. DialogueRNN [9], introduces a more sophisticated architecture that tracks the emotional states of individual speakers over time. It comprises three GRUs: one for modeling the global conversational context, another for capturing speaker-specific states, and a third for maintaining the emotional dynamics. This design enables the model to effectively handle multiparty conversations and to adapt to the emotional shifts of each participant.

These models collectively highlight the progression from basic contextual modeling to more intricate architectures that account for the complexities of human emotional exchanges in conversations. Their contributions have laid the groundwork for developing more nuanced and responsive ERC systems.

Following these earlier models, recent developments have shifted toward transformer-based architectures to more effectively handle the complexities of multimodal emotion recognition in conversation.

Our approach builds on the Self-Distillation Transformer (SDT) [8], a transformer-based model designed for multimodal ERC. SDT integrates intra- and inter-modal Transformers, along with positional and speaker embeddings, and employs a hierarchical gated fusion strategy to dynamically combine multimodal features. A key innovation is its self-distillation mechanism, where the model uses its own fused predictions as supervision to enhance unimodal learning. This strategy eliminates the need for a separate teacher model while benefiting from both soft and hard label supervision, ultimately improving generalization.

3 Method

We are going to present an overview of how the SDT model works and outlines its main components, including feature extraction, multimodal fusion, and the self-distillation process. Note that the original model have also a video modality that we decide to remove due to the expensive computational cost. A figure illustrating the model architecture can be found in the Appendix A.1.

3.1 Feature Extraction

The SDT model starts by extracting features from the textual and acoustic modalities of each utterance in a conversation. Textual features are obtained using a fine-tuned RoBERTa [7] model, where the [CLS] token embedding represents the semantic content of the utterance. Acoustic features are extracted using the openSMILE [2] toolkit, which captures prosodic and paralinguistic aspects such as pitch, energy, and intonation. These modality-specific features are then projected into a shared embedding space using a 1D convolutional layer to ensure compatibility for subsequent processing.

3.2 Modality Encoder

After feature extraction, the modality encoder models both intra-modal and inter-modal dependencies.

- **Intra-modal transformers** capture sequential patterns within each modality independently, allowing the model to understand how emotions evolve over time in the same channel.
- **Inter-modal transformers** enable cross-modal interactions by letting one modality (e.g., text) attend to another (e.g., audio).

To enrich these representations, positional embeddings are added to encode the order of utterances, and speaker embeddings are included to distinguish between different speakers, helping the model track emotional dynamics across turns.

3.3 Hierarchical Gated Fusion

To combine information from the two modalities, the model employs a hierarchical gated fusion mechanism. At the first level, each modality selectively filters information received from the other through a gating mechanism, enhancing relevant content and suppressing noise. At the second level, the model computes a weighted combination of the enhanced text and audio representations using a softmax gate, dynamically adjusting the importance of each modality per utterance. This results in a fused representation that reflects the most informative aspects of both channels.

3.4 Self-distillation

Knowledge distillation is a learning paradigm originally proposed for model compression [4], where a compact "student" model is trained to replicate the behavior of a larger, more accurate "teacher" model. Instead of training only on ground-truth labels (hard targets), the student also learns from the soft output probabilities of the teacher, which capture richer information about class distributions and inter-class similarities. This process has been extended beyond model compression to improve performance, generalization, and robustness of neural networks. Building on this idea, the SDT framework incorporates a self-distillation strategy to improve the learning of individual modalities. During training, the full model acts as a teacher, producing both hard targets and soft targets. Each individual modality serves as a student and learns to predict both the hard and soft labels. The loss function combines cross-entropy loss with the ground-truth labels and KL-divergence loss with the soft predictions. This training scheme helps each modality benefit from the overall model's understanding, leading to more discriminative and consistent representations, even in the absence of the other modality.

4 Experimental analysis

This section describes the experiments that we have conducted to evaluate our implementation. First, Subsection 4.1 introduces the dataset used for training and evaluation. Subsection 4.2 outlines the preprocessing pipeline. Then, Subsection 4.3 describes our self-distillation framework and the custom loss functions used. Subsection 4.4 introduces the integration of speaker personality embeddings. Subsection 4.5 describes the experimental setup, and finally, Subsection 4.6 reports the results, including comparison with the original model, three baselines and an ablation study.

4.1 Dataset

We conducted our evaluations on the **IEMOCAP dataset** [1], a widely used benchmark for multimodal emotion recognition in conversations. The dataset comprises approximately 12 hours of audiovisual recordings of dyadic interactions between professional actors, involving both scripted and improvised scenarios. Each interaction is segmented into utterances and annotated with categorical emotion labels by multiple human annotators. The dataset provides rich multimodal information, including speech (audio), transcripts (text), and video (visual facial expressions), making it highly suitable for research in emotion recognition. We opted to exclude visual features in our experiments due to computational constraints and a focus on exploring the interaction between the textual and audio modalities. Consequently, only the text transcripts and audio features were used as input for the model, along with their multimodal fusion.

Dataset Structure and Split

The IEMOCAP dataset is organized into five sessions, each featuring a conversation between a **male** and a **female** actor. To ensure speaker independence between training and testing, we adopt a session-based split strategy:

- Sessions 1 – 4 are used for training and validation.
- Session 5 is reserved exclusively for testing.

Within Sessions 1 – 4, we further apply a random 80%/20% split to create the training and validation sets. This approach avoids speaker overlap between sets, thereby reducing data leakage and allowing for a more robust evaluation of model generalization.

While the dataset originally includes ten emotion categories — *happy*, *sad*, *angry*, *neutral*, *frustrated*, *excited*, *surprised*, *fearful*, *disgusted*, and *other* — many prior studies adopt a reduced set to improve class balance and simplify classification. We group certain emotions to define a six-class setting:

- *Happy* and *Excited* $\rightarrow$ **Happy**
- *Frustrated*, *Surprised*, *Fearful*, *Disgusted* $\rightarrow$ **Other**

4.2 Preprocessing Pipeline

We preprocess both the textual and audio modalities of the dataset. Each session is handled independently to ensure consistency and modularity across our feature extraction pipeline.

To manage the computational cost of preprocessing, we serialize the processed data using Python's `Pickle` module. This allows us to cache the outputs and avoid repeating expensive computations during every training run, thereby improving both efficiency and reproducibility.

Each conversation is processed and stored in a structured format. Specifically, we organize the data at the conversation level, where each conversation is represented as a dictionary containing:

- **Speaker metadata** – such as speaker ID and gender.
- **Emotion labels** – the categorical target emotion for each utterance.
- **Textual features** – extracted using a pretrained RoBERTa model.
- **Audio features** – extracted using the OpenSMILE toolkit.

Additionally, we maintain two arrays to store the conversation IDs, enabling a clear separation of training, validation, and test sets during downstream processing.

To leverage the serialized data, we implement a PyTorch `Dataset` class that reconstructs each conversation's structure. A validation split is derived from the training set, and we build three `DataLoader` objects—for training, validation, and testing—used throughout the training and evaluation of our models.

We extract textual and acoustic features using RoBERTa and OpenSMILE respectively. For RoBERTa, we explore both the base pretrained model and a fine-tuned variant to assess the benefits of task adaptation. In our multimodal model, RoBERTa is used to generate feature embeddings from each utterance. To obtain more accurate embeddings, we fine-tune the `roberta-base` model for emotion classification using only the textual modality from IEMOCAP.

For this fine-tuning process, we create a dedicated dataset stored in a `csv` file, structured as Table 1 shows.

Table 1: Structure of the fine-tuning dataset

Column Name	Type	Description
utterance_id	string	The ID of the utterance in IEMOCAP
text	string	The utterance transcript
label	int	Numerical encoding of the label

We split this dataset using stratified sampling into training (80%) and test (20%) sets to preserve the class distribution. Each sample is tokenized using the `RobertaTokenizer` from Hugging Face, with padding and truncation applied to a fixed maximum length. The classification model is a `RobertaForSequenceClassification` initialized with a number of output labels equal to the distinct emotion classes.

We manage the training via the Hugging Face `Trainer` API trying different configurations:

- Learning rate: `2e-5`, `2e-3`
- Batch size: `8`, `16`

- Epochs: `3`, `8`, `10`, `13`
- Weight decay: `0.01`, `0.05`, `0.001`

After training, the model is evaluated on the test set and metrics are computed using `scikit-learn`, including precision, recall, and F1-score. Although we experimented with multiple parameter settings, the performance remained low (accuracy below 50%). Consequently, we adopted **Emoberta** [6], a pretrained version of RoBERTa fine-tuned for ERC on IEMOCAP and MELD datasets. This model captures conversational context and significantly improves the quality of the textual embeddings for our use case.

For the audio modality, the SDT model uses **OpenSMILE** to extract acoustic feature embeddings. OpenSMILE is a signal processing toolkit, so fine-tuning is not applicable. Nevertheless, we evaluated various predefined feature sets offered by OpenSMILE to improve performance. The configurations tested are summarized in Table 2.

Table 2: OpenSmile configurations

FeatureSet	Description	Recommended Use
`ComParE_2016`	Very large (6373 dims), used for emotion/speaker/age/health recognition	General use, powerful but high-dimensional
`emobase`	988 dims, classic emotion feature set from INTERSPEECH 2009	Compact alternative for emotion recognition
`eGeMAPSv02`	Minimalistic (88–100 dims), designed for interpretability	Best for low-resource or interpretable setups

We ultimately decided against using `ComParE_2016` due to its high dimensionality and computational cost. Instead, we selected a more compact yet expressive set suitable for our computational constraints. The next section details the performance results achieved under each configuration introduced in the preprocessing stage.

4.3 Self-Distillation with Custom Loss Functions

To effectively leverage the multimodal nature of ERC, we adopt a **self-distillation** framework. In this setup, the model uses its own fused multimodal predictions to supervise its unimodal branches. This approach removes the need for a teacher model while still leveraging the benefits of distillation.

4.3.1 Custom Loss Functions

To guide the training process of our model, we implement two custom loss functions: the **Masked Negative Log-Likelihood Loss** for direct supervision, and the **Masked Kullback-Leibler Divergence Loss** for knowledge distillation.

i. Masked Negative Log-Likelihood Loss:

This loss is a standard supervised classification objective. It is applied independently to the log-softmax outputs of the text, audio, and multimodal classifiers. To ensure that padded utterances do not contribute to the loss, we apply a binary mask that zeroes out their effect. The loss is computed as follows:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{\sum_i m_i} \sum_{i=1}^{N} m_i \cdot (-\log p_{i,y_i}) \tag{1}$$

where N is the number of utterances; p_{i,y_i} is the predicted probability (after log-softmax) for the correct class y_i of utterance i; $m_i \in \{0, 1\}$ is the mask indicating whether the i-th utterance is valid (not padding).

ii. Masked KL Divergence Loss:

This distillation loss aligns the unimodal classifier predictions with those of the multimodal classifier. Both student and teacher outputs are softened using a temperature parameter $\tau > 0$, which helps the student model learn from the teacher's full probability distribution rather than just the most confident class.

The loss is computed as:

$$\mathcal{L}_{\text{KL}} = \frac{\tau^2}{\sum_i m_i} \sum_{i=1}^{N} m_i \cdot \sum_{c=1}^{C} q_{i,c}^{(\tau)} \log \frac{q_{i,c}^{(\tau)}}{p_{i,c}^{(\tau)}} \tag{2}$$

where $q_{i,c}^{(\tau)} = \text{softmax}(z_i^{\text{teacher}}/\tau)$ is the teacher's softened probability for class c; $p_{i,c}^{(\tau)} = \text{softmax}(z_i^{\text{student}}/\tau)$ is the student's softened probability; z_i^{teacher} and z_i^{student} are the logits of the teacher and student models for utterance i; m_i is the mask value for utterance i; C is the number of classes.

This loss encourages unimodal branches to mimic the richer distribution learned by the multimodal branch, improving both unimodal performance and the coherence of the overall multimodal system.

Total Loss:

The final objective combines these losses across all modalities:

$$\mathcal{L} = \gamma_1 \mathcal{L}_{\text{task}} + \gamma_2 \mathcal{L}_{\text{CE}} + \gamma_3 \mathcal{L}_{\text{KL}} \tag{3}$$

$$\mathcal{L}_{\text{CE}} = \sum_{m \in \{t,a\}} \mathcal{L}_{\text{CE}}^{m}, \quad \mathcal{L}_{\text{KL}} = \sum_{m \in \{t,a\}} \mathcal{L}_{\text{KL}}^{m} \tag{4}$$

where γ_1, γ_2, γ_3 are weighting coefficients controlling the contribution of task loss, cross-entropy loss on individual modalities, and KL divergence loss respectively.

This formulation lets each modality learn from both ground truth and multimodal fusion, enhancing emotional representation without requiring external teacher models.

4.4 Incorporating Speaker Personality Embeddings

In ERC, speaker personality can significantly influence emotional expression, while the original SDT model uses only speaker ID embeddings. Inspired by BiosERC [12], we enrich the SDT model by integrating speaker personality information. To achieve this, we first use a large language model—specifically **ChatGPT-4**—to generate natural language summaries that describe each speaker's personality based on their conversational behavior (see Appendix A.2). These descriptions are then encoded into dense vectors using **RoBERTa**. Finally the resulting personality embeddings are injected into the SDT model by augmenting the original speaker embeddings.

Formally, the original speaker embedding component SE in the SDT model is modified from:

$$H_m = U'_m + PE + SE \tag{5}$$

to

$$SE_{\text{augmented}} = V_s(s_j) + W \cdot h_{desc_j} \tag{6}$$

$$H_m = U'_m + PE + SE_{\text{augmented}} \tag{7}$$

where $V_s(s_j)$ is the original trainable speaker ID embedding for speaker s_j; h_descj is the RoBERTa-encoded vector of the LLM-generated personality description for speaker s_j; W is a learnable linear transformation projecting h_descj to the same dimension as $V_s(s_j)$.

This augmentation allows personality traits to influence all intra- and inter-modal transformer layers, enabling the model to interpret emotions in a more personalized and context-aware manner.

4.5 Experimental settings

In our experiments, we designed a comprehensive setup to train and fine-tune the SDT for the task of ERC. Our setup encompasses two main phases: model selection via hyperparameter tuning and final model training using the best configuration.

Our experiments were conducted on a 13th Gen Intel® Core™ i7-1355U processor running at 1.70 GHz, with the exception of the audio-only experiments, which were carried out on a system equipped with a GeForce RTX 3050 GPU.

We evaluated several model configurations to analyze the individual and joint contributions of the different modalities and the self-distillation mechanism. These included the full SDT model using both text and audio with self-distillation, text-only and audio-only versions with the mechanism applied individually, and cross-modal setups where one modality (text-only, audio-only) is used for classification while both are supervised, allowing us to test whether knowledge transfer improves performance. We also tested a full model without self-distillation to isolate its effect. Through these configurations, we aim to investigate the importance of each modality, the role of self-distillation, and how their interactions affect the overall performance in ERC.

4.5.1 Model Selection and Hyperparameter Tuning

To identify the optimal configuration of the full SDT model, we performed a grid search separately for each OpenSmile feature set (eGeMAPSv02, emobase). Each combination was evaluated by training the model and computing its F1-score on the validation set, with the grid search exhaustively exploring all possible parameter configurations. Throughout this process, several settings were kept fixed: the loss weighting coefficients were set to $\gamma_1 = 1.0$, $\gamma_2 = 1.0$, and $\gamma_3 = 1.0$; the model was trained for 100 epochs; the number of attention heads was set to 8; and the batch size was fixed at 16.

The best hyperparameter configuration for each feature set, based on validation performance, is reported in Table 3.

Table 3: Best hyperparameter configurations for different OpenSmile configuration

Modality	τ	**learning rate**	**dropout**	**weight decay**	**model dim**
eGeMAPSv02	2.0	0.0005	0.005	0.005	16
emobase	1.5	0.008	0.005	0.005	32

Based on the initial grid search results, we selected the **eGeMAPSv02** configuration for final training, as it yielded the best performance. To further refine the model, we conducted a second grid search focusing on a narrower range of hyperparameters[1], the final configuration selected after this second grid search is showed in Table 4.

Table 4: Best hyperparameter configurations for the final SDT model

Model	τ	**learning rate**	**dropout**	**weight decay**	**model dim**	**batch size**	**n heads**
full SDT	0.7	0.0005	0.01	0.01	144	32	18

The final SDT model was trained using this best configuration, with the aim of evaluating performance using both hard labels (cross-entropy) and soft targets (KL divergence) derived from the fused teacher prediction, in accordance with the self-distillation strategy.

Importantly, we reused this best-found configuration to evaluate the full model without self-distillation, enabling a fair comparison and isolating the impact of the distillation mechanism. This configuration was also used for the variant that integrates speaker personality embeddings.

For the text-only and audio-only variants, we conducted separate grid searches to determine the best hyperparameters tailored to each setting, as these unimodal configurations differ significantly in capacity and modality complexity. For consistency, we used the same OpenSmile configuration selected for the final training of the full SDT model.

To identify the optimal configuration for the text-only and audio-only without self-distillation, we conducted two separate grid searches[2]. Each configuration was evaluated based on its F1-score on the validation set to select the best-performing model.

[1]The sets of hyperparameters tested is showed in Table 12

[2]Sets of tested hyperparameters are showed respectively in Table 13 and Table 14

For the text-only and audio-only models with self-distillation, we adopted the same optimal hyperparameter configurations identified for their counterparts without self-distillation. Based on a limited set of exploratory experiments, we set the temperature parameter to $\tau = 3.5$ for the audio modality, and $\tau = 4.0$ for the text modality.

As for the full SDT model, all the other parameters were kept fixed, with the same values showed before; and the best-performing hyperparameter configuration for each setting was selected based on validation F1-score. Table 5 reports the chosen value for each experiment.

Table 5: Best hyperparameter configurations for unimodal variants (with and without self-distillation)

Modality	**learning rate**	**dropout**	**weight decay**	**model dim**	**batch size**	**n heads**
text-only	0.008	0.005	0.01	48	128	16
audio-only	0.005	0.005	0.0005	32	32	16

4.6 Results

In the following, we present the results of the experiments outlined in the previous section, reporting the performance values obtained on the test set. The training performance values are reported in Appendix A.3.

In Table 6 are reported the result achived using the full SDT model with different OpenSmile configuration and the parameters showed in Table 3.

Table 6: Different OpenSmile configuration

Configuration	**ACC**	**F1-score**
eGeMAPSv02a	**48.99**	**34.21**
emobase	47.51	30.61

Table 7 presents the results achieved by our reimplementation of the SDT model, which utilizes a different combination of modalities, alongside the results reported in the original paper for comparison; the hyperparameter are the one reported in Table 4.

Table 7: Performance comparison of SDT and its ablations

	ORIGINAL		**OUR**	
	ACC	**F1-score**	**ACC**	**F1-score**
Modality				
Text + Audio	**72.52**	**72.75**	58.94	45.26
Text only	66.42	66.58	59.80	43.88
Audio only	59.77	59.34	58.25	42.93
Self-distillation				
w/o self-distillation	70.73	71.10	56.36	41.39

Figure 1 illustrates the evolution of the total loss during training and test of the final SDT model.

4.6.1 Baseline comparison

To evaluate the effectiveness of our reimplemented SDT model, we compared its performance against several widely used baseline models for emotion recognition in conversations—CMN, ICON, and DialogueRNN—using results obtained on the IEMOCAP dataset. Additionally, we include a comparison with the original SDT model, which incorporates all three modalities: textual, acoustic, and visual.

It is important to highlight that all of the above models utilize textual, acoustic, and visual features for multimodal representation. In contrast, our SDT implementation only integrates textual and acoustic modalities. The comparison of results is presented in Table 8.

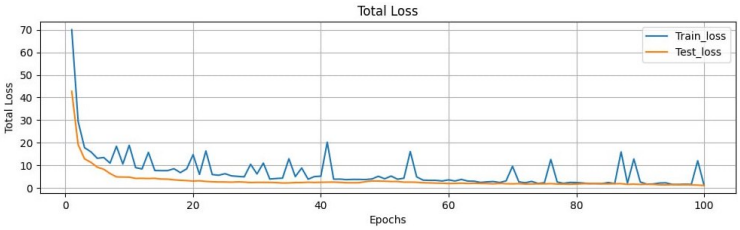

Figure 1: Trend of the total loss during training and testing. The training curve shows some instability, likely due to the imbalanced label distribution in the dataset.

Table 8: Comparison between different baseline models

Models	Accuracy	F1-score
SDT (our)	**58.94**	**45.26**
SDT (original)	73.95	74.08
CMN	56.87	56.33
ICON	62.85	62.25
DialogueRNN	65.43	64.29

4.6.2 Model Variant Evaluation

We further investigated how the self-distillation mechanism affects unimodal setups by evaluating SDT using only textual or audio features, while still applying self-distillation across both modalities. This design aims to determine whether the distillation process can transfer knowledge from the complementary modality (e.g. audio to text) and enhance performance in a unimodal setting. The results of this experiment are reported in Table 9. The parameter settings employed in this experiment are summarized in Table 5.

Table 9: Performance of SDT in unimodal settings (text-only and audio-only) with cross-modal self-distillation applied

	ACC	F1-score
text-only (w/ self-distillation)	56.31	41.20
audio-only (w/ self-distillation)	60.00	47.56

Finally, we evaluated the impact of integrating speaker personality embeddings into the SDT model. Table 10 reports the results of this experiment.

Table 10: Performance comparison of SDT with speaker personality embeddings

Modality	Accuracy	F1-score
SDT	58.94	45.26
SDT + bios	**59.26**	**45.96**

For each experiment, we visualized the total loss across training epochs to monitor convergence behavior. For the complete SDT model, we also present plots of accuracy and F1-score over time, along with a confusion matrix to evaluate class-wise prediction performance. All visualizations are available in Appendix A.5.

5 Discussion

The results obtained in our reimplementation and evaluation of the SDT model offer several insights into the robustness and flexibility of multimodal emotion recognition architectures. While we followed the general structure proposed by Ma et al. [8], our implementation introduced a number of variations driven by computational constraints and exploratory choices that diverge from the original

configuration.We deviated from the original SDT model's hyperparameters and architecture to better suit our computational constraints, using validation-based tuning instead of fixed settings. Key changes include replacing RoBERTa with EmoBERTa, simplifying acoustic features, and adjusting convolutional dimensions. Additionally, our model has a significantly smaller overall size, and we modified several parameters to strike a balance between efficiency and performance.

The full model does not match the result obtained by [8], particularly when fusing both text and audio modalities. However, However, our unimodal results (Table 7) are competitive. This suggests that with more careful tuning of the fusion hyperparameters, our model could potentially improve in the multimodal setting.

Table 7 also displays the result obtained cutting out the self-distillation mechanism, that shows lower performance with respect to our full model, proving that this mechanism is beneficial to the model. Although this result is not directly comparable with the original implementation due to differences in modalities used (we exclude the video stream), it still highlights the importance of self-distillation in our architecture.

Further, as part of our model variant evaluation, we compare our SDT model with an implementation that includes speaker biography embeddings, that should help the model to discern the different emotions. As shown in Table 10, this enhancement yields a modest improvement in both accuracy and F1-score, suggesting that integrating speaker-specific traits may further strengthen the model's emotional understanding.

Table 8 compares different baselines architectures. Our implementation outperforms one of the early baselines, CMN, in terms of accuracy, and is competitive with ICON and DialogueRNN—especially considering it relies on fewer modalities (text and audio only, as opposed to the full multimodal input used in the baselines). This suggests that our reimplementation, although not matching the full potential of the original SDT, still captures valuable cross-modal emotional patterns and remains effective within a constrained modality setup.

As shown in Table 11, self-distillation significantly enhances the generalization of the audio-only model, evidenced by a higher test F1-score despite lower performance on the training set. For the text-only model, the impact is more mixed, with a slight drop in test accuracy and F1-score. Overall, self-distillation appears to be more beneficial for the weaker modality (audio) by enhancing robustness and generalization, whereas its effect on the stronger modality (text) is more nuanced and may depend on additional tuning.

6 Conclusions

In conclusion, our reimplementation of the SDT model demonstrates that effective emotion recognition from text and audio can still be achieved under limited computational resources and time constraints.

Despite not having access to powerful machines or extensive training time, we were able to reproduce a competitive architecture, introducing key adaptations such as reduced model complexity, alternative pretrained embeddings, and simplified acoustic features. While our full multimodal performance lags behind the original SDT, the unimodal results and the improvements brought by self-distillation and speaker personality embeddings affirm the strength and flexibility of the approach. These findings highlight the potential for building lightweight yet effective emotion recognition systems, and pave the way for future improvements through targeted tuning and broader modality integration when resources allow.

With additional resources, several promising directions could be explored to further improve performance. First, incorporating visual features alongside text and audio could enhance the model's multimodal understanding. Second, fine-tuning RoBERTa specifically for the task of emotion recognition in conversations may yield more task-adapted textual representations. For the BiosERC component, future work could involve exploring different LLMs for speaker descriptions, refining their encoding, and replacing static embeddings with attention mechanisms for dynamic personality modeling.

The source code is available here.

References

[1] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359, 2008.

[2] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462, 2010.

[3] Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2018, page 2122, 2018.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[5] Jingwen Hu, Yuchen Liu, Jinming Zhao, and Qin Jin. MMGCN: Multimodal fusion via deep graph convolution network for emotion recognition in conversation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5666–5675, Online, August 2021. Association for Computational Linguistics.

[6] Taewoon Kim and Piek Vossen. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *arXiv preprint arXiv:2108.12009*, 2021.

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

[8] Hui Ma, Jian Wang, Hongfei Lin, Bo Zhang, Yijia Zhang, and Bo Xu. A transformer-based model with self-distillation for multimodal emotion recognition in conversations. *IEEE Transactions on Multimedia*, 2023.

[9] Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6818–6825, 2019.

[10] Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE access*, 7:100943–100953, 2019.

[11] Tao Shi and Shao-Lun Huang. MultiEMO: An attention-based correlation-aware multimodal fusion framework for emotion recognition in conversations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14752–14766, Toronto, Canada, July 2023. Association for Computational Linguistics.

[12] Jieying Xue, Minh-Phuong Nguyen, Blake Matheny, and Le-Minh Nguyen. Bioserc: Integrating biography speakers supported by llms for erc tasks. In *International Conference on Artificial Neural Networks*, pages 277–292. Springer, 2024.

A Appendix

A.1 Model structure

Figure 2 shows the overall structure of the SDT model that we reimplemented, without the use of video modality.

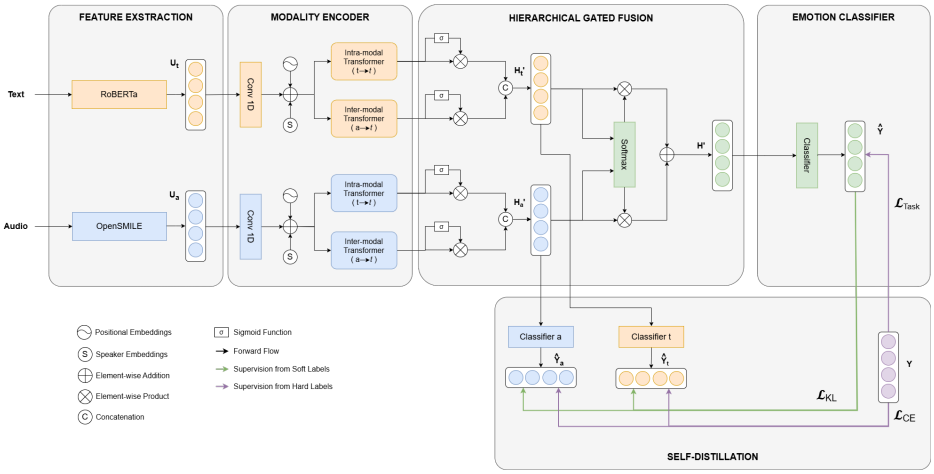

Figure 2: Architecture of the reimplemented SDT model

A.2 Prompt

To ensure consistency and relevance, we designed a specific prompt that guided the model to focus on traits observable from dialogue patterns, emotional tone, and interaction style.

```
Given this conversation between speakers:

{conversation}

In overall above conversation, what do you think about the characteristics of
speaker {speaker_name}?

(Note:  provide an answer within 250 words)
```

A.3 Results

Table 11: Performance of SDT in unimodal settings (text-only and audio-only) with cross-modal self-distillation applied.

	Train		Test	
	ACC	**F1-score**	**ACC**	**F1-score**
OpenSmile configuration				
eGeMAPSv02a	40.66	28.9	**48.99**	**34.21**
emobase	46.26	30.8	47.51	30.61
Modality				
SDT (Text + Audio)	40.35	33.04	58.94	45.26
Text only	66.64	53.30	59.08	43.88
Audio only	53.07	36.8	58.25	42.93
SDT + bios	41.28	35.11	**59.26**	**45.96**
Self-distillation				
w/o self-distillation	30.97	31.46	56.36	41.39
text-only (w/ self-distillation)	61.80	52.56	56.31	41.20
audio-only (w/ self-distillation)	48.11	35.88	**60.00**	47.56

A.4 Hyperparameters

Table 12: The table shows the different sets of hyperparameters tested to find the best settings for the full SDT model.

Hyperparameter	First grid search	Second grid search
Temperature (τ)	[1.0, 1.5, 2.0]	[1.0, 1.5, 2.0, 2.5, 3.0]
Learning rate	[0.001, 0.005, 0.008]	[0.0001, 0.0005, 0.0008, 0.001, 0.005, 0.008]
Dropout rate	[0.001, 0.005]	[0.001, 0.005, 0.008, 0.01]
Weight decay	[0.001, 0.005]	[0.001, 0.005, 0.008, 0.01]
Model dimension	[16, 32]	—
Batch size	—	[16, 32]

Table 13: The table shows the sets of hyperparameters tested to find the best settings for the text-only model.

Hyperparameter	Values
Learning rate	[0.0001, 0.0005, 0.001, 0.005, 0.008]
Dropout rate	[0.0001, 0.0005, 0.001, 0.005]
Weight decay	[0.0001, 0.0005, 0.001, 0.005]
Model dimension	[32, 40, 48]
Batch size	[32, 64, 128]
Number of Heads	[8, 10, 16]

Table 14: The table shows the sets of hyperparameters tested to find the best settings for the audio-only model.

Hyperparameter	Values
Learning rate	[0.0001, 0.0005, 0.001, 0.005]
Dropout rate	[0.0001, 0.0005, 0.001, 0.005]
Weight decay	[0.0001, 0.0005, 0.001, 0.005]
Model dimension	[32, 40, 128, 512]
Batch size	[16, 32]
Number of Heads	[8, 16]

A.5 Plot

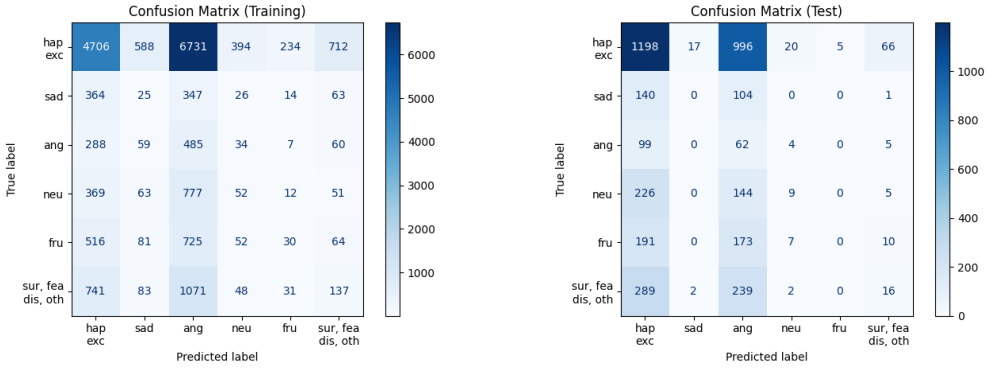

Figure 3: Confusion Matrix for Train and Test of the full SDT model.

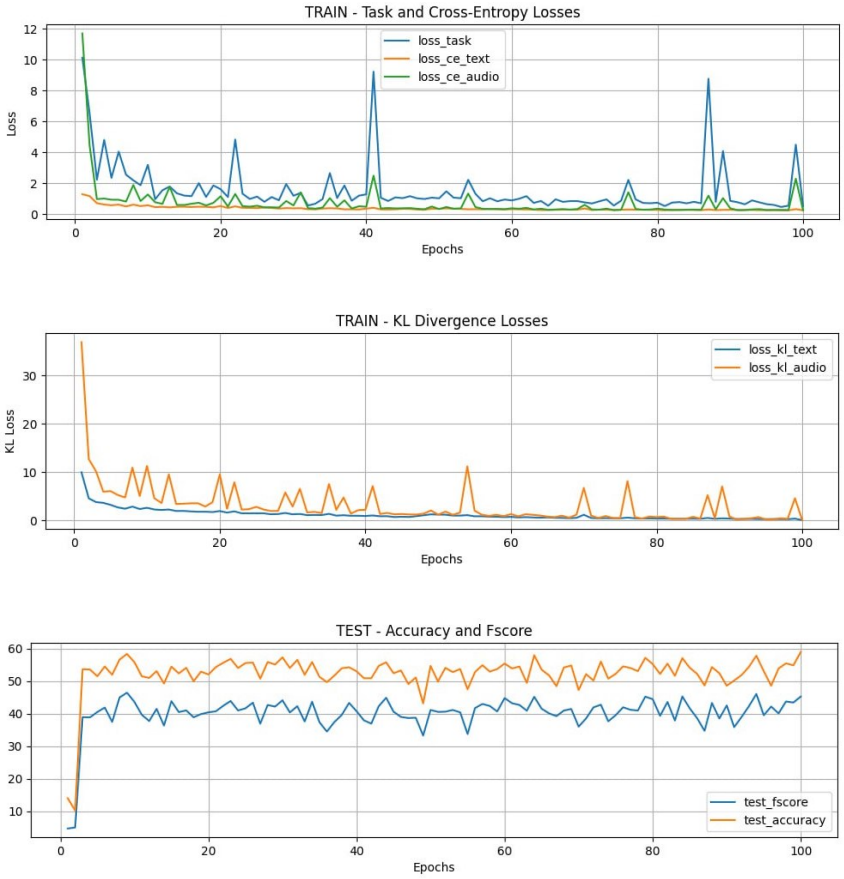

Figure 4: Visualization of SDT model performance: (top) Cross-Entropy loss components, (middle) KL-divergence loss components, and (bottom) test accuracy and F1-score across epochs.

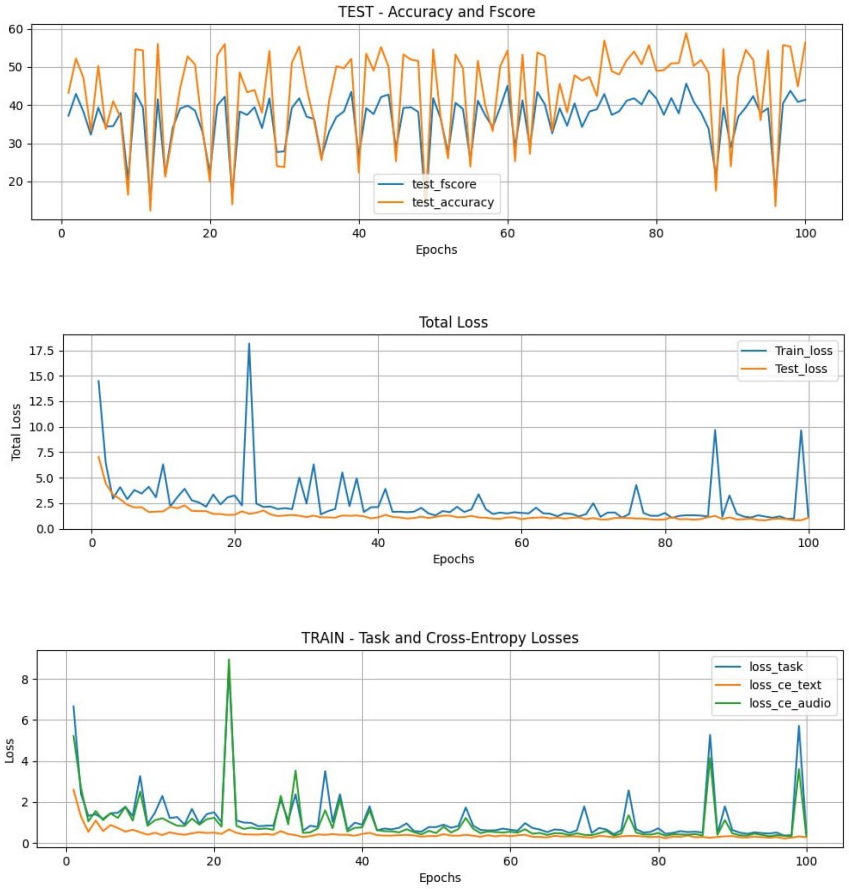

Figure 5: Performance of the SDT model without self-distillation: (top) Accuracy and F1-score through epochs, (middle) total loss components, and (bottom) cross-entropy loss components.

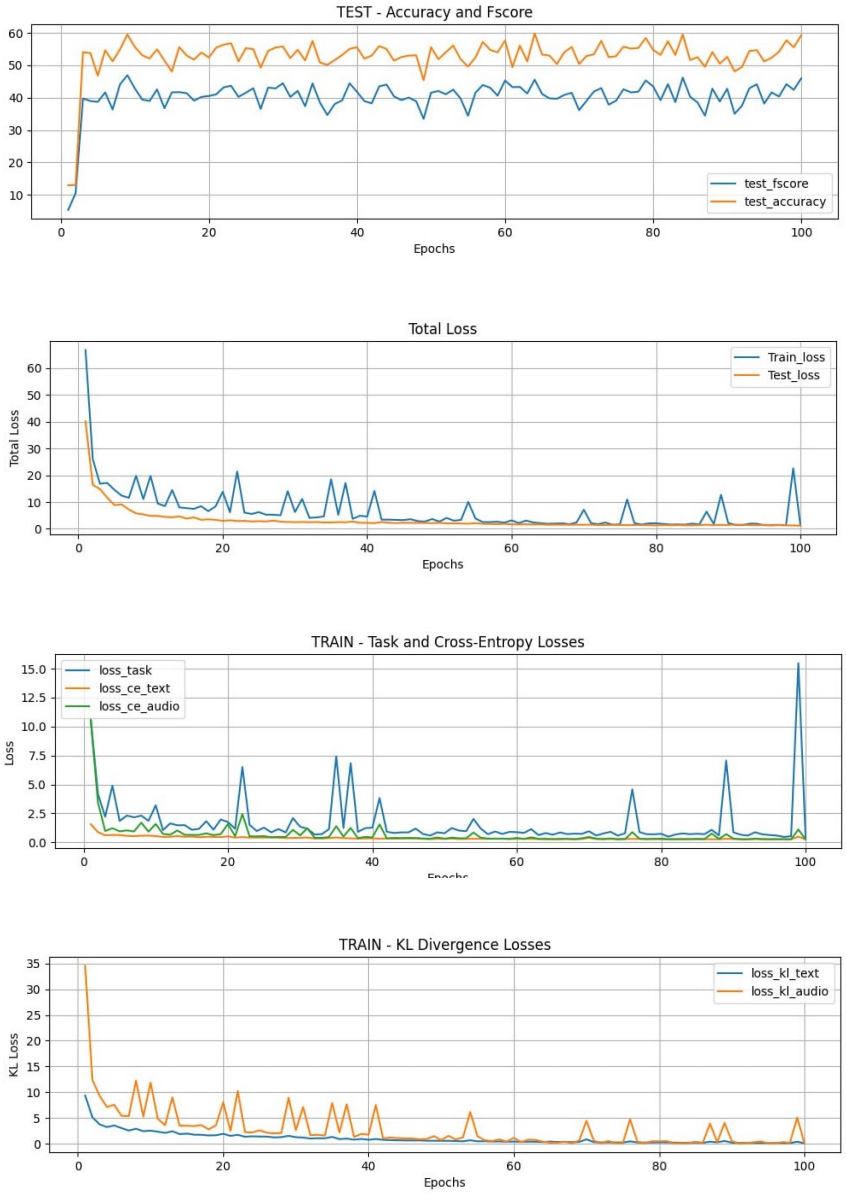

Figure 6: SDT model with speaker personality embedding: (top) Accuracy and F1-score through epochs, (second) overall loss components, (third) cross-entropy loss components, and (bottom) KL loss components.

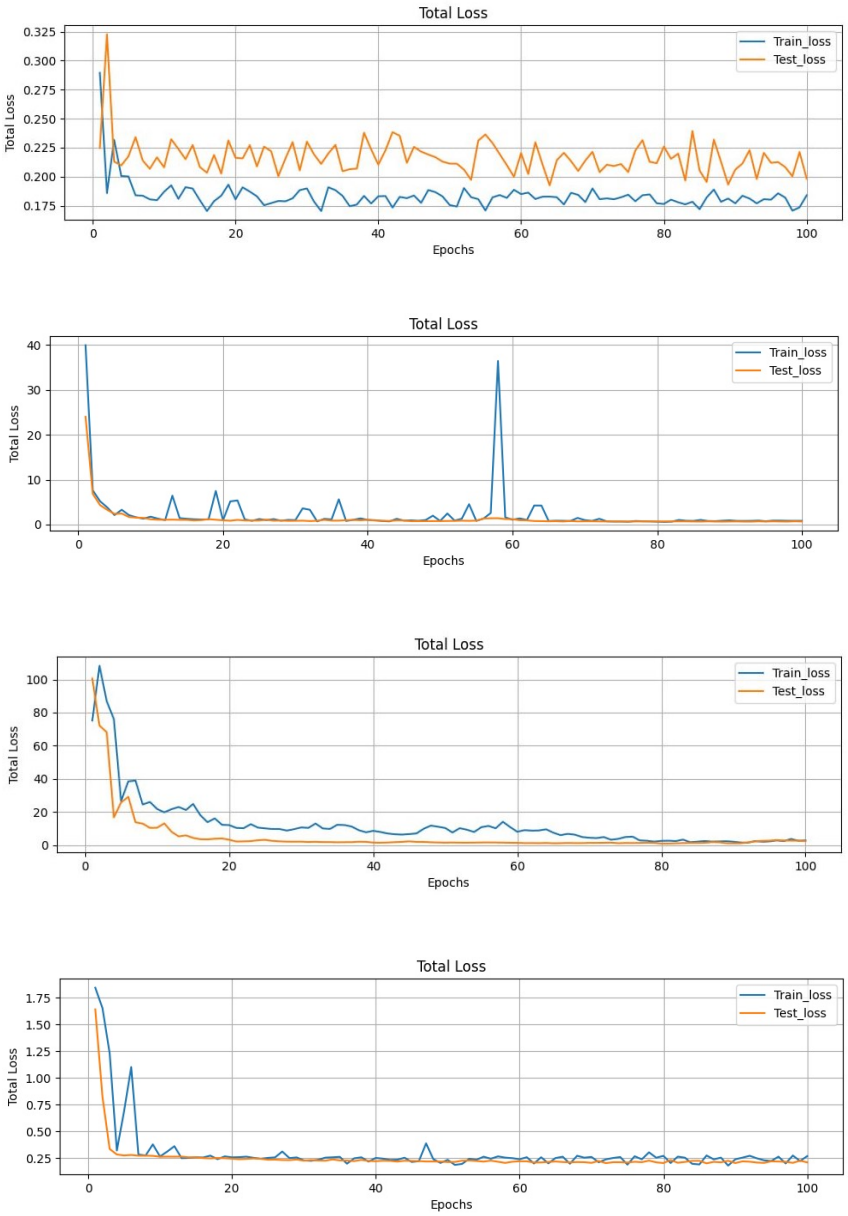

Figure 7: Total loss comparison for unimodal SDT models: (top) text-only without self-distillation, (second) audio-only without self-distillation, (third) text-only with self-distillation, and (bottom) audio-only with self-distillation.