M2-Mixer: A Multimodal Mixer with Multi-head Loss for Classification from Multimodal Data

Grigor Bezirganyan
Aix-Marseille Univ, CNRS, LIS
Marseille, France
grigor.bezirganyan@etu.univ-amu.fr

Sana Sellami
Aix Marseille Univ CNRS, LIS
Marseille, France
sana.sellami@univ-amu.fr

Laure Berti-Équille
IRD, ESPACE-DEV
Montpellier, France
laure.berti@ird.fr

Sébastien Fournier
Aix Marseille Univ CNRS, LIS
Marseille, France
sebastien.fournier@univ-amu.fr

2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386252

***Abstract*—In this paper, we propose M2-Mixer, an MLP-Mixer based architecture with multi-head loss for multimodal classification. It achieves better performances than the convolutional, recurrent, or neural architecture search based baseline models with the main advantage of conceptual and computational simplicity. The proposed multi-head loss function addresses the problem of modality predominance (i.e., when one of the modalities is favored over the others by the training algorithm). Our experiments demonstrate that our multimodal mixer architecture, combined with the multi-head loss function, outperforms the baseline models on two benchmark multimodal datasets: AV-MNIST and MIMIC-III with respectively, on average, $+0.43\%$ in accuracy and 6.4 times reduction in training time and $+0.33\%$ in accuracy and 13.3 times reduction in training time, compared with previous best performing models.**

***Index Terms*—Multimodal Deep Learning, Multimodal Data Fusion, Multimodal Classification, MLP, MLP-mixer**

I. Introduction

In recent years, multi-modal deep learning has gained significant popularity due to its ability to fuse data from multiple sources, resulting in a more comprehensive and accurate representation of the real world [2]. This approach is widely used in various domains, including healthcare, finance, multimedia applications, natural language processing, and others. Multimodal deep classifiers use data from different modalities, such as audio, video, text, image, and time series to improve the predictive accuracy by leveraging the diversity of data.

Although multimodal deep learning has proven to work well on many tasks and datasets, current state-of-the-art methods suffer from limitations that must be addressed. First, there is a trade-off between the performance and computational and conceptual simplicity in multimodal networks [10]. In other words, complex models that can achieve good performance are often computationally expensive. The current state-of-the-art performances are achieved by complex and computationally costly methods, such as neural architecture search based models [18, 30] or large transformer-based models [16, 29]. Additionally, some approaches require pre-trained models for feature extraction from modalities [11, 16]. This kind of pre-trained models may not always exist, and pre-training them will take additional effort and computational resources. Therefore, there is a need for approaches that are conceptually and computationally simple but still perform comparably or better than the more complex models.

Another limitation of multimodal deep learning models is that often the joined learning strategy may favor one modality over the other, and as a result, find suboptimal representations for the modalities. As Wang et al. [26] showed, multimodal models with a joined training strategy can even have lower performance compared to uni-modal architectures. In their paper, the authors propose a gradient blending approach that blends gradients according to a heuristic that takes into account the over-fitting and generalization of the network. However, MultiBench, the multimodal benchmark [10] showed that not only the gradient blending approach is much slower in training time than standard networks of similar size, but it also does not generalize well on other datasets.

In this paper, we aim to address both of these limitations by proposing a computationally and conceptually simple architecture that achieves comparable or better performances than other existing models and reduces the optimization imbalance. Our contributions are as follows.

(1) We propose a novel extension of MLP-Mixers [23] to multimodal scenarios. While convolutions and neural architecture search can achieve decent performance, we show that they are not necessary: an "all-MLP" (i.e., network composed only of Multi-Layer Perceptrons), multimodal network can achieve competitive results at a cheaper cost. Because the foundational elements of the architecture are simple matrix multiplication and scalar non-linearity function, it is conceptually simpler and easier to design and debug. The all-MLP nature of the proposed architecture reduces the computational costs. Furthermore, the modularity of our architecture enables us to easily use and combine other MLP-based methods without painful modification and code rewriting.

(2) We combine our multimodal approach, M2-Mixer with a multi-head loss function to reduce the optimization imbalance problem described by Wang et al. [26]. We show that this outperforms all the baseline models while keeping the performance time low. The multi-head loss, being a simplified version of Gradient Blend [26] shows that a simple loss weighting can still achieve good performance, without the huge time overhead introduced by the complex weight calculation of Gradient Blend.

(3) We perform extensive experiments on two benchmark mul-

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.

timodal datasets, AV-MNIST [25] and MIMIC-III [7] used in the literature. Our experiments demonstrate that the M2-Mixer architecture outperforms the baseline architectures, including the Gradient Blending [26] and MFAS approaches [18], both in time (on average 6.4 times on AV-MNIST and 13.3 times on MIMIC-III) and accuracy (on average $+0.43\%$ on AV-MNIST and $+0.33\%$ on MIMIC-III).

This paper is structured as follows. Section II provides an overview of the related work in multimodal deep learning and MLP-mixers. Section III describes our approach called M2-Mixer for multimodal classification. Section IV presents the experimental results and a comparison with baseline models. Finally, section V discusses our contributions, the limitations of our approach and provides a conclusion.

II. Related Work

Multimodal Deep Learning aims to utilize information from different types of data (e.g., image, video, audio, time series, text, etc.) to enhance the accuracy of predictors. The design of multimodal deep learning methods requires many design choices and tests of configurations. Given a multimodal dataset and a prediction task, one has to choose the appropriate paradigm, the feature extraction methods, the fusion function, etc. Recently, neural architecture search (NAS) methods have been proposed to find the optimal architecture given a limited search space (e.g., MFAS [18]). These methods achieved state-of-the-art results on several multimodal datasets used as benchmarks. They outperformed the scores of ad-hoc models with a substantially longer training time [10]. Liang et al. [10] showed that while MFAS [18] achieves better performance on the datasets it was designed for, it does not generalize well to other datasets. Additionally, Wang et al. [26] showed that, in a multimodal setting, there is an optimization imbalance problem, where one modality is favored over the others. Their solution however introduces additional complexity to the training process to find the weights for blending the gradient of each modality. With M2-Mixer, we propose a simple ad-hoc alternative that can outperform complex models (including NAS), while keeping the computation costs at a minimum. To address the optimization imbalance problem, we opt for a simple weighting approach, which still achieves good results, but reduces the computational overhead. Our approach is similar to the multi-task loss introduced by Vielzeuf et al. [25] where they optimize the sum of losses for each separate modality and the multimodal head. However, in addition to the uniform weights, we suggest and test the effects of gradually increasing the importance of the multimodal loss.

MLP-Mixers and Multimodal MLPs. MLP-Mixers were proposed by Tolstikhin et al. [23] as a simple all-MLP alternative to convolutional networks and transformers for computer vision tasks. They demonstrated that convolutions and transformers are not necessary to achieve decent performances. MLP-Mixers work by dividing images into patches, flattening the patches, and then applying MLPs on token dimension and channel dimension. They are theoretically simple and

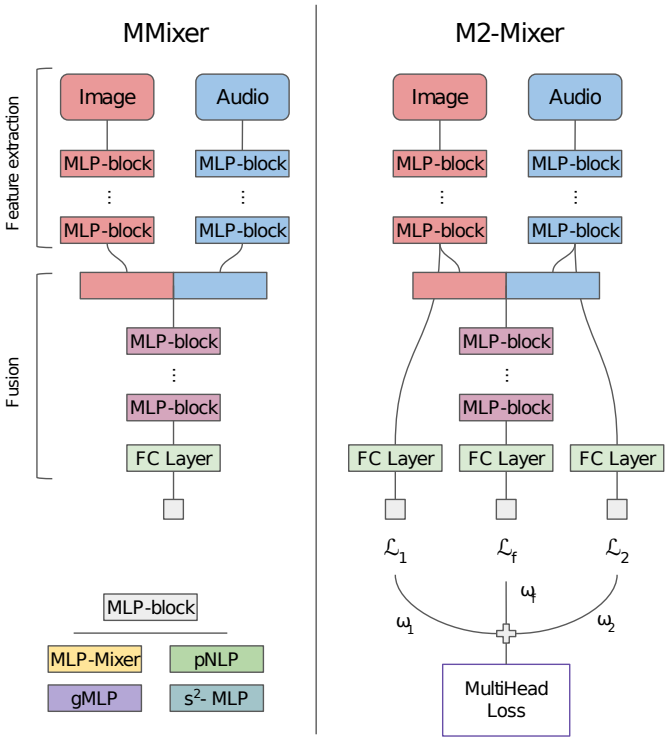

Fig. 1: The proposed architectures of MMixer and M2-Mixer for two modalities. MLP-blocks serve as feature extractors. Then concatenated features are fused using additional MLP-blocks. Finally, a Global Average Pooling layer is used, and the resulting representations are fed into a task-specific head(s). For M2-Mixer, the MultiHead Loss is the weighted sum of the modality losses.

computationally efficient architectures that achieve competitive results against state-of-the-art models for computer vision tasks. A new line of research dedicated to MLP-based networks emerged and researchers concentrate on extending the architecture for different types of data, and improving its performance [3, 14, 31]. Recent works extend MLP mixers to the multimodal domain [12, 21, 28]. Nevertheless, they use other convolutional or recurrent networks or transformers for feature extraction and then use MLP-mixers on top of those extracted features. To the best of our knowledge, our approach is the first all-MLP architecture, where MLPs are used from feature extraction to classification from multimodal data.

III. Our Approach: Multimodal Mixers

By leveraging the simplicity/performance trade-off of MLP-Mixers and incorporating multimodal fusion techniques, we aim to develop a simple yet effective architecture that can handle the complexity of multimodal data while maintaining high classification performance with leveraging data diversity. We extend the idea of MLP-Mixers to create an all-MLP architecture for a multimodal setting. The all-MLP architecture entails that we will only make use of matrix multiplications, transformations, and scalar non-linearities. For the sake of simplicity, we will assume that we have a bi-modal input,

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.

as it is straightforward to extend the bi-modal case to many modalities.

To build feature extractors from the modalities, we define the basic building blocks that we call *MLP-Blocks*. Let $x_i \in \mathbb{R}^{S\times P\times C}$ be the input to the i-th modality where S is the batch size, P is the number of patches, and C is the number of channels. The MLP-block, noted b is defined as an all-MLP function $b : \mathbb{R}^{S\times P\times C} \to \mathbb{R}^{S\times P\times C}$, that receives an input x_i and returns an output, noted θ of the same size as in Eq. 1.

$$\theta_i = b_i(x_i) \text{ with } x_i, \theta_i \in \mathbb{R}^{S\times P\times C}. \tag{1}$$

MLP-blocks can be stacked onto each other to build deeper architectures as defined in Eq. 2:

$$\Theta_i = B_i^n(x_i) = b_i^n \circ b_i^{n-1} \circ \cdots \circ b_i^2 \circ b_i^1(x_i), \tag{2}$$

where N is the number of MLP-blocks ($n \in \{1 \dots N\}$). These blocks provide us with an interface to create MLP-based multimodal architectures in a modular manner. Since each block accepts and returns the same type of data, one can combine and switch different types of MLP-blocks to create the most optimal architecture. While, in this paper, we use an MLP-Mixer [23] as MLP blocks, one can choose between blocks such as S^2-MLP [31], pNLP-Mixer [3], gMLP [14], etc., depending on the requirements, the modality types, and dataset characteristics.

Since we want to have an all-MLP architecture, we will use MLPs to fuse the features Θ_i extracted from the modalities. We concatenate the feature matrices into one matrix, and forward it into another stack of MLP-Blocks as presented in Eq. 3 with $n_f \in \{1 \dots N_f\}, n_1 \in \{1 \dots N_1\}, n_2 \in \{1 \dots N_2\}$:

$$\Theta_f = B_f^{n_f}\left(B_1^{n_1}(x_1); B_2^{n_2}(x_2)\right) = B_n^{n_f}\left(\Theta_1; \Theta_2\right). \tag{3}$$

This fusion MLP-Block B_f aims to learn the cross-modal relationships between modalities. This is where the Mixer architecture shines. Since it applies Feed-Forward networks in both: patch and channel dimensions, the token-mixing MLPs can learn which features to take from which modality and from which patch.

Finally, we need to use a classifier head on top of the learned multimodal features. One approach could be to flatten the features and provide to the fully connected classification layers. However, the number of parameters in that case will be relatively high, and the network will be more prone to over-fit. Instead, similarly to MLP-Mixer architecture, we use global average pooling (GAP) [13] to reduce the number of parameters, computational time and avoid over-fitting at this level. We then add a classification head on top as formalized in Eq. 4:

$$\hat{y} = \arg\max_c \sigma\left(\text{GAP}\left(\Theta_f\right)\right), \tag{4}$$

where σ is the Sigmoid function in the case of binary classification, and the Softmax function for multi-class classification. The general architecture of the Multimodal Mixer is depicted in Fig. 1.

In our experiments, we noticed that the feature extractors of the network are not optimally trained during the end-to-end joined training. This is a known issue in the literature, and several approaches attempt to solve it in different ways [4, 17, 26]. We address this issue in our architecture called M2-Mixer (Multi-head Multimodal Mixer), where we assign task heads to the modalities in addition to the task head on top of the fused features (similarly to Gradient Blending [26] but without expensive computations), and we get three branches as defined in Eq. 5 (shown in Fig. 1 for 2 modalities).

$$\hat{y}_i = \arg\max_c \sigma\left(\text{GAP}\left(\Theta_i\right)\right), \quad i \in \{f, 1, 2\}. \tag{5}$$

We then proceed to minimize the summed Multi-head loss as defined in Eq. 6:

$$\begin{aligned} &\mathcal{L}(\hat{y}_f, \hat{y}_1, \hat{y}_2, y) = w_f\mathcal{L}_f(\hat{y}_f, y) + w_1\mathcal{L}_1(\hat{y}_1, y) + w_2\mathcal{L}_2(\hat{y}_2, y) \\ &\text{with } \sum_i w_i = 1 \end{aligned} \tag{6}$$

where w_f, w_1, and w_2 are weighting factors that control how strong the optimization on the specific branch must be. When $w_f = 1$, the architecture is equivalent to the standard non-multi-head multimodal mixer. This loss function governs how much each branch shall be optimized. The loss modality encoders are now forced to learn optimal representations by the joined objective function.

In the Gradient Blending approach, one must train the network to completion on a subset of data [26] to find the optimal weights for the per-modality loss functions, which is a very costly task. Furthermore, as shown by Liang et al. [10] and our experiments (see Section IV), Gradient Blend can not always find the optimal weights and sometimes reduces the model's accuracy. While uniform weights may not be the optimal weights for the loss function, they can dramatically reduce the computational overhead and do not have the generalization issue of Gradient Blend.

Since the final result comes from the multi-modal branch, it makes sense to gradually increase its importance with a piece-wise linear function defined in Eq. 7:

$$\begin{aligned} w_f^i &= \begin{cases} \min(w_f^i + (i-k)\cdot\eta, 1), & \text{if } i > k \\ w_f^i, & \text{otherwise} \end{cases} \\ w_1^i &= w_2^i = \frac{1}{2}\left(1 - w_f^i\right), \quad i \in \{1 \dots E\}, \end{aligned} \tag{7}$$

where η is the rate of weight change in each epoch, k is the epoch after which the weights will start to change, and E is the total number of epochs. Here, η and k are hyper-parameters to optimize.

IV. Experiments

In this section, we compare the results of our architecture of Multimodal Mixers and its configurations against **nine** baseline methods. For all the experiments, we used the same machine in an internal cluster with an Intel(R) Xeon(R) Silver 4114 @ 2.20GHz CPU and an NVIDIA GeForce RTX 2080 Ti GPU. Around 1,400 hours of computation time was used

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.

to conduct all the experiments necessary for this project. For reproducing our results, including our Multimodal Mixers, you can find the source code and publicly available datasets at `https://github.com/bezirganyan/m2-mixer`.

A. Datasets and Baselines

We chose AV-MNIST and MIMIC-III datasets for our experiments. They are from different disciplines and provide different types of modalities (image, audio, time series, tabular data). Furthermore, they are medium-sized, which enabled us to perform extensive experimentation for architecture design and comparison.

AV-MNIST [25] is a multimodal dataset created by combining the handwritten image recognition MNIST dataset [8], with the Tidigits [9] digit pronunciation dataset. Since current unimodal models already have close to perfect performance on these datasets, the authors reduce the information from each modality by removing 75% of energy from the MNIST dataset by PCA, and adding random noise samples from ESC-50 [19] dataset to the audio modality. Instead of Tidigits, we use the FSDD dataset [5] and follow the codes by a third-party repository[1] for data generation and pre-processing. The audio files are transformed into 112×112 spectrograms, while the image modality remains a collection of 28×28 images.

MIMIC-III [7] is a comprehensive, freely available database that contains de-identified health-related data of over 40,000 patients who received critical care at the Beth Israel Deaconess Medical Center between 2001 and 2012. In our experiments, we closely follow the data pre-processing and splitting procedure described in MultiBench paper [10]. The patient data is organized into two modalities: time-series modality, which consists of 12 different medical measurements of the patients taken each hour for 24 hours (a matrix of size 24×5); and static modality, which is vector of size 5, representing various medical information about the patient. These modalities are used for mortality prediction, i.e., to predict if the patient will die in 1 day, 2 days, 3 days, 1 week, 1 year, or after 1 year (6 classes).

Baselines: We use the following nine architectures as baslines from MultiBench [10] benchmark:

• **Simple Late Fusion** uses a fully connected network on top of concatenated feature extractors for classification.

• **Low-rank Tensor Fusion** [15] uses low-rank weight tensors to improve model efficiency, without compromising its performance. The LRTF manages to scale linearly with the number of modalities.

• **MFAS** (Multimodal Fusion Architecture Search) [18] is using a neural architecture search to find an optimal architecture given the search space and the search algorithm for multimodal datasets.

• **ReFNet** [20] combines the fusion network with a decoding / defusing module, which ensures that both unimodal and fused representations are strongly encoded in the latent fusion space.

[1]https://github.com/slyviacassell/_MFAS

• **MVAE** [27] uses a product-of-expert inference and a sub-sampled training paradigm to solve the multi-modal inference problem.

• **MFM** (Multimodal Factorization Model) [24] tries to learn improved multimodal representations by factorizing them into multimodal discriminative factors that are shared between modalities and modality-specific generative factors.

• **CCA** [22] tries to learn the correlation between modalities using a deep canonical correlation analysis [1].

• **MI-Matrix** [6] aims to generalize tensor products to include learnable parameters that capture the interactions between streams of information from the different modalities.

• **GradBlend** (Gradient Blending) [26] aims to analyze the over-fitting of different modalities to find out the optimal blending strategy.

B. Evaluation setup

We propose 6 configurations for our approach that differ in terms of the number of parameters and the dataset they are designed for: MMixer, M2-Mixer_B, M2-Mixer_M, M2-Mixer_S, M2-Mixer_Hm M2-Mixer_LC. MMixer is our suggested all-MLP approach, without Multi-head loss. The goal is to demonstrate that MMixer can achieve competitive performance with only simple operations. Next, we propose 3 configurations of M2-Mixers for AV-MNIST dataset (B-Big, M-Medium, and S-Small in terms of the number of parameters), and 2 configurations for MIMIC-III (H-Healthcare, LC-Linear Change with uniform and linearly changing weights respectively). All experiments mentioned in the paper are run 10 times with different random seeds, and the average scores with their standard deviation are reported.

For all of our experiments on the AV-MNIST dataset, we use a starting learning rate of 0.01, with a scheduler that reduces the learning rate by a factor of 10 if there is no improvement in the validation loss for 2 epochs. For MIMIC-III we use the same starting learning rate, but we reduce it every 5 epochs if there is no improvement in the validation loss. These numbers are found by empirical tests and by observing how the validation loss function behaves. For the detailed configuration of the models and all hyper-parameters, please refer to our code base.

C. Experimental Results

Results on AV-MNIST. We represent the audio modality as spectrograms and use MLP-Mixers as feature extractors for both modalities. In Table I, we present the results of our extensive experiments conducted on the AV-MNIST dataset. Our approach, MMixer, represents an original multimodal mixer architecture without the Multi-head loss. Despite the absence of the Multi-head loss, MMixer demonstrates comparable performance to the other baseline models in terms of accuracy and testing time, with the exception of the MFAS model. Notably, MFAS, which incorporates an architecture search, outperforms all other baseline architectures, and also MMixer. However, it is important to acknowledge that MFAS exhibits considerably longer training and testing times compared to the

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Results on **AV-MNIST** dataset.

Architecture	Acc. (%) (avg)	Acc. (%) (max)	F-1 (%) (avg)	Training Time (s)	Test Time (s)	Training Params (M)
Simple Late Fusion [10]	71.55 ± 0.6	72.30	70.91 ± 0.7	873 ± 5	5.5 ± 0.1	0.26
LRTF [15]	71.53 ± 0.4	72.30	70.98 ± 0.4	1108 ± 30	5 ± 0.2	0.25
MFAS [18]	72.63 ± 0.2	72.92	72.08 ± 0.2	65710 ± 12817	12.3 ± 1	0.14 [2]
RefNet [20]	71.60 ± 0.8	72.75	70.79 ± 0.9	891 ± 5	7.5 ± 5	14.1
MVAE [27]	70.72 ± 1.2	72.43	69.99 ± 1.0	1129 ± 12	5.6 ± 0.2	0.81
MFM [24]	71.40 ± 0.8	72.28	70.76 ± 0.9	1110 ± 5	5.4 ± 0.13	0.92
CCA [22]	72.01 ± 0.4	72.76	71.42 ± 0.5	$\mathbf{378 \pm 1}$	$\mathbf{3.48 \pm 0.07}$	0.25
MI-Matrix [6]	71.14 ± 0.4	71.83	70.45 ± 0.5	631 ± 2	5.5 ± 0.2	2.53
GradBlend [26]	68.71 ± 0.7	69.51	68.52 ± 1.3	43768 ± 5554	8.1 ± 2	0.29
MMixer	71.58 ± 0.6	72.66	71.10 ± 0.6	1632 ± 158	5.3 ± 0.25	0.88
M2-Mixer_B	$\mathbf{73.06 \pm 0.2}$	**73.34**	$\mathbf{72.45 \pm 0.2}$	10271 ± 6578	5.72 ± 0.35	8.3
M2-Mixer_M	72.81 ± 0.2	73.20	72.27 ± 0.2	4147 ± 1642	4.67 ± 0.24	0.88
M2-Mixer_S	71.99 ± 0.37	72.49	71.46 ± 0.4	4688 ± 2182	4.516 ± 0.34	0.19

TABLE II: Results on **MIMIC-III** dataset.

Architecture	Acc. (%) (avg)	Acc. (%) (max)	F-1 (%) (avg)	Training Time (s)	Test Time (s)	Training Params (M)
Simple Late Fusion [10]	77.78 ± 0.4	78.23	77.78 ± 0.4	78 ± 0.43	0.52 ± 0.04	0.034
LRTF [15]	76.82 ± 0.9	78.33	76.82 ± 0.9	729 ± 11.23	0.62 ± 0.05	0.008
MFAS [18]	78.02 ± 0.4	78.63	78.02 ± 0.4	8043 ± 633	1.24 ± 0.12	0.086 [2]
MVAE [27]	78.1 ± 0.2	78.45	78.10 ± 0.2	659 ± 1.62	0.76 ± 0.04	0.312
MFM [24]	77.97 ± 0.5	78.51	77.97 ± 0.5	213 ± 3.21	0.63 ± 0.05	0.323
Ml-Matrix [6]	77.67 ± 0.4	78.2	77.67 ± 0.4	95 ± 0.52	0.58 ± 0.03	0.801
GradBlend [26]	78.1 ± 0.3	78.51	78.10 ± 0.3	7988 ± 239	0.66 ± 0.03	0.063
M2-Mixer_H	78.32 ± 0.3	**79.03**	78.32 ± 0.3	840 ± 119	0.32 ± 0.14	0.029
M2-Mixer_LC	$\mathbf{78.43 \pm 0.3}$	78.76	$\mathbf{78.43 \pm 0.3}$	597 ± 113	0.32 ± 0.13	0.029

other architectures. Consequently, our objective is to surpass the performance of MFAS while maintaining efficient train and test durations.

To design our model, we decided to check the contributions of individual modalities and determine if the optimization imbalance problem mentioned in [26] is also present in our architecture (i.e., if modality encoders are optimally trained by optimizing the uni-modal encoder). First, to understand the upper bound of the accuracy that modality encoders can achieve, we train them separately, on AV-MNIST dataset. As we can see in Table III (Uni-modal models) the feature encoder on image modality can achieve an accuracy of 39.93% and on audio modality, an accuracy of 66.10%. To assess the optimal training of the modality encoders, we isolated the feature encoders, froze their weights, and trained a linear classifier layer on top of these fixed encoders. The corresponding metrics can be found in Table III, in the "MMixer" column. Upon examination, we observed that the accuracy of the audio modality encoder was approximately 4.35% lower compared to the accuracy of the respective uni-modal model. Similarly, the image modality encoder achieved an accuracy approximately 2.15% lower than its corresponding uni-modal model. These results indicate that the encoders were indeed not optimally trained by the joined (single-loss) strategy. In contrast, when we analyze the uni-modal accuracy values of the M2-Mixer_M model (which is identical to MMixer except for the multi-head loss), we find that the performance closely aligns with the results obtained by our uni-modal models trained independently (in the column Uni-modal models in Table III). This finding highlights the beneficial impact of the multi-head loss in getting the optimal representations for the modality encoders. Wang et al. [26] solved the issue by attaching a task head to each of the modalities and minimizing the weighted loss, where the weights are determined by expensive computations. We can see in Table I that their solution, Gradient Blending, is more than 40 times slower than the next slowest architecture (MVAE) in the baseline list. Moreover, Gradient Blending fails to compete with other models in terms of accuracy, achieving sub-optimal accuracy of only 68.71% on average. Furthermore, we can see that our uniform weighting approach increases the performance from 71.58% to 72.81% for the medium model (M2-Mixer_M)

TABLE III: Accuracies (%) of feature extractors of different models on AV-MNIST dataset

Modality	Uni-modal models	MMixer	M2-Mixer_M
Audio	39.93	35.58	39.90
Image	66.10	63.95	66.10

[2] The number of input parameters of MFAS at the start of training. More parameters are generated during the architecture search process.

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.

with approximately the same number of parameters. M2-Mixer_M also outperforms the previous best MFAS model, while being around 16 times faster in training time, and around 2.6 times faster in testing time. We also design and test two other configurations of M2-Mixer: a bigger model, named M2-Mixer_B, with around 10 times more parameters, and a much smaller model, M2-Mixer_S. From Table I, we can see that M2-Mixer_B achieves even better performance (averaged accuracy of 73.06%), nevertheless we lose in training time around 2.5 times. The smaller model, while being much lighter with fewer parameters, is not faster than the medium model. This is because, while each epoch takes less time, it takes more epochs for the model to converge to the optimal representation.

Results on MIMIC-III. Next, we test the architecture on the MIMIC-III dataset. The challenging parts of adapting the model to another dataset are the differences in the types of modalities and size of the dataset. Two modalities are included in MIMIC-III dataset: time-series modality and static modality (i.e., a vector of size 5). While MLP-Mixers were originally designed for vision tasks, we can apply MLPs on two dimensions and adapt the model to different types of data. In our specific case, each sample in the time-series modality is a matrix of size 24×5. Each line of the matrix (i.e., each time point) is similar to the flattened patches of the original MLP-mixer architecture. Hence, we can skip the patching phase and apply the remaining part of the MLP-Mixer architecture to the modality. For the static modality, we only have vectors of size 5, and we cannot really fit MLP-Mixers on this modality. Hence, we use a standard MLP as an MLP block for this modality.

The next challenge is the relatively small size of the dataset (80 MB) whereas MLP mixers are designed preferably for large-scale datasets; their performance increases with bigger models and datasets [23]. In our experiments, though, the M2-Mixer with uniform weights (M2-Mixer_H in Table II) achieved a decent performance, slightly outperforming MVAE and Gradient Blending. Since our main objective is to get a good performance on the multimodal branch, we tried to start training with uniform weights, but slightly increase the importance of the multimodal branch according to the Eq. 7. Through experiments we found the optimal parameters for this dataset to be $\eta = 0.05$ and $k = 1$. This further improved the accuracy to 78.43% (M2-Mixer_LC in Table II). However, as stated before, linear change brings additional hyper-parameters to the model, and there may be additional costs to find the optimal ones.

V. Conclusion

In this paper, we present M2-Mixer: an all-MLP based architecture that outperforms the baseline models while being computationally more effective and conceptually simple. An important advantage of our abstract definition of MLP-blocks is that it brings modularity to the architecture: depending to the use case, each block can be easily replaced by another MLP-block. For example, if one needs to deal with textual data input, they can use pNLP-Mixer [3] as a block, or one can use S^2-MLPs to improve the performance on smaller datasets. Moreover, the research in all-MLP architectures dramatically increased after the introduction of MLP-Mixers. Over time, the cost of such architectures may decrease while achieving better performances. In this case, one will only need to swap the MLP-blocks with the newest blocks and benefit from the improvements and latest advances in this domain. Despite its strengths, our architecture does have some limitations that pave the way for future research. For one, while we successfully developed and tested our architecture on bi-modal data, scaling it on-demand to multiple modalities may be challenging due to two key factors. Firstly, the use of concatenation as a fusion function may not be the ideal case for many modalities, despite its effectiveness with bi-modal data. Secondly, our multi-head loss function may be computationally demanding when it is used with a large number of task heads. This could potentially slow down the optimization process and impact the overall performance of the model. Additionally, optimizing the loss function when it consists of many individual losses can also pose a challenge, as we are not sure how the optimization algorithm will behave in this scenario. In our experiments, we used 2 pairs of 2 different types of modality (audio/image and time series/static data), but it is crucial to test our architecture with larger datasets combining 3 or more types of modality (e.g., text/image/audio, etc.). Finally, in addition to adjusting the weights based on Eq. 7, it is necessary to conduct additional experiments to determine the optimal values for the parameters η and k for our model, which requires more computation time and resources.

Overall, we believe that our work represents a significant step forward in the development of all-MLP based architectures for multi-modal and multi-task learning, with an exciting potential for future applications.

References

[1] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep Canonical Correlation Analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255. PMLR, May 2013.

[2] K. Bayoudh, R. Knani, F. Hamdaoui, and A. Mtibaa. A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The Visual Computer*, 38(8):2939–2970, Aug. 2022.

[3] F. Fusco, D. Pascual, P. W. J. Staar, and D. Antognini. pnlp-mixer: an efficient all-mlp architecture for language. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023*, pages 53–60, 2023.

[4] Y. Huang, J. Lin, C. Zhou, H. Yang, and L. Huang. Modality Competition: What Makes Joint Training of Multi-modal Network Fail in Deep Learning? (Provably). In *Proceedings of the 39th International Conference on Machine Learning*, pages 9226–9259. PMLR, June 2022.

[5] Z. Jackson, C. Souza, J. Flaks, Y. Pan, H. Nicolas, and A. Thite. Jakobovski/free-spoken-digit-dataset: v1.0.8, Aug. 2018.

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.

[6] S. M. Jayakumar, W. M. Czarnecki, J. Menick, J. Schwarz, J. W. Rae, S. Osindero, Y. W. Teh, T. Harley, and R. Pascanu. Multiplicative interactions and where to find them. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[7] A. Johnson, T. Pollard, and R. Mark. MIMIC-III Clinical Database, 2015.

[8] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.

[9] R. G. Leonard and G. R. Doddington. TIDIGITS, 1993.

[10] P. P. Liang, Y. Lyu, X. Fan, Z. Wu, Y. Cheng, J. Wu, L. Chen, P. Wu, M. A. Lee, Y. Zhu, R. Salakhutdinov, and L. Morency. Multibench: Multiscale benchmarks for multimodal representation learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*, 2021.

[11] T. Liang, G. Lin, M. Wan, T. Li, G. Ma, and F. Lv. Expanding Large Pre-trained Unimodal Models with Multimodal Information Injection for Image-Text Multimodal Classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15471–15480. IEEE, June 2022.

[12] H. Lin, P. Zhang, J. Ling, Z. Yang, L. K. Lee, and W. Liu. PS-Mixer: A Polar-Vector and Strength-Vector Mixer Model for Multimodal Sentiment Analysis. *Information Processing & Management*, 60(2):103229, Mar. 2023.

[13] M. Lin, Q. Chen, and S. Yan. Network in network. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[14] H. Liu, Z. Dai, D. R. So, and Q. V. Le. Pay attention to mlps. In *Advances in Neural Information Processing Systems 34*, pages 9204–9215, 2021.

[15] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Bagher Zadeh, and L.-P. Morency. Efficient Low-rank Multimodal Fusion With Modality-Specific Factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2247–2256, July 2018.

[16] K. Miyazawa, Y. Kyuragi, and T. Nagai. Simple and Effective Multimodal Learning Based on Pre-Trained Transformer Models. *IEEE Access*, 10:29821–29833, 2022.

[17] X. Peng, Y. Wei, A. Deng, D. Wang, and D. Hu. Balanced Multimodal Learning via On-the-fly Gradient Modulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8237. IEEE, June 2022.

[18] J. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie. MFAS: multimodal fusion architecture search. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6966–6975. Computer Vision Foundation / IEEE, 2019.

[19] K. J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[20] S. Sankaran, D. Yang, and S.-N. Lim. Multimodal Fusion Refiner Networks, Apr. 2021. arXiv:2104.03435 [cs], unpublished.

[21] H. Sun, H. Wang, J. Liu, Y. Chen, and L. Lin. Cubemlp: An mlp-based model for multimodal sentiment analysis and depression estimation. In *MM '22: The 30th ACM International Conference on Multimedia*, pages 3722–3729. ACM, 2022.

[22] Z. Sun, P. K. Sarma, W. A. Sethares, and Y. Liang. Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8992–8999. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6431.

[23] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems 34*, pages 24261–24272, 2021.

[24] Y. H. Tsai, P. P. Liang, A. Zadeh, L. Morency, and R. Salakhutdinov. Learning factorized multimodal representations. In *7th International Conference on Learning Representations, ICLR*, 2019.

[25] V. Vielzeuf, A. Lechervy, S. Pateux, and F. Jurie. Centralnet: A multilayer approach for multimodal fusion. In *Computer Vision - ECCV 2018 Workshops, Proceedings, Part VI*, volume 11134 of *Lecture Notes in Computer Science*, pages 575–589. Springer, 2018.

[26] W. Wang, D. Tran, and M. Feiszli. What makes training multi-modal classification networks hard? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020*, pages 12692–12702. Computer Vision Foundation / IEEE, 2020.

[27] M. Wu and N. D. Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems 31*, pages 5580–5590, 2018.

[28] J. Xia, M. Zhuge, T. Geng, S. Fan, Y. Wei, Z. He, and F. Zheng. Skating-mixer: Long-term sport audio-visual modeling with mlps. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*, pages 2901–2909. AAAI Press, 2023.

[29] P. Xu, X. Zhu, and D. A. Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:12113–12132, 2022.

[30] Z. Xu, D. R. So, and A. M. Dai. MUFASA: multimodal fusion architecture search for electronic health records. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 10532–10540. AAAI Press, 2021.

[31] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li. S^2-mlp: Spatial-shift MLP architecture for vision. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV*, pages 3615–3624. IEEE, 2022.

Authorized licensed use limited to: Laure BERTI. Downloaded on April 24,2025 at 07:38:22 UTC from IEEE Xplore. Restrictions apply.