Appendix

Contents

A Broader Impact Statement **24**

B Background: Multimodal Representation Learning **26**

B.1 Problem Statement . . . 26

B.2 Technical Challenges . . . 26

C MULTIBENCH Datasets **28**

C.1 Dataset Selection . . . 28

C.2 Dataset Details . . . 29

C.2.1 Affective Computing . . . 29

C.2.2 Healthcare . . . 32

C.2.3 Robotics . . . 33

C.2.4 Finance . . . 35

C.2.5 HCI . . . 36

C.2.6 Multimedia . . . 37

C.3 Documentation . . . 40

C.4 Benchmark Distribution . . . 48

C.5 Hosting and Maintenance . . . 48

C.6 Author Statement . . . 48

C.7 License . . . 48

C.8 Metadata . . . 48

C.9 Persistence of MULTIBENCH . . . 49

D MULTIBENCH Evaluation Protocol **50**

D.1 Performance . . . 50

D.2 Complexity . . . 50

D.3 Robustness to Imperfect Data . . . 50

D.3.1 Modality-specific Imperfections . . . 50

D.3.2 Multimodal Imperfections . . . 52

D.3.3 Robustness Measure . . . 53

D.4 Aggregating Measures Across Datasets and Tasks . . . 54

E MULTIZOO: A Zoo of Multimodal Algorithms **55**

E.1 Selection of Algorithms in MULTIZOO . . . 55

E.2 Data Preprocessing . . . 55

E.3 Fusion Paradigms . . . 56

E.4 Optimization Objectives . . . 57

E.5 Training Procedures . . . 58

E.6 Domain-specific Methods . . . 58

F Integrating MULTIBENCH and MULTIZOO: A Brief Tutorial 59

F.1 Reading the Dataset . . . 59
F.2 Unimodal Models . . . 59
F.3 Multimodal Models . . . 60
F.4 Classification Head . . . 60
F.5 Optimization Objectives . . . 60
F.6 Training Structures . . . 60
F.7 Performance Evaluation . . . 60
F.8 Complexity Evaluation . . . 61
F.9 Robustness Evaluation . . . 61
F.10 Code Snippets . . . 61

G Experimental Setup 62

G.1 Affective Computing . . . 62
G.2 Healthcare . . . 63
G.3 Robotics . . . 63
G.4 Finance . . . 63
G.5 HCI . . . 63
G.6 Multimedia . . . 63

H Experimental Results 71

H.1 Affective Computing . . . 71
H.2 Healthcare . . . 75
H.3 Robotics . . . 77
H.4 Finance . . . 80
H.5 HCI . . . 83
H.6 Multimedia . . . 85
H.7 Performance . . . 88
H.8 Complexity . . . 90
H.9 Robustness . . . 92
H.10 Summary of Takeaway Messages . . . 93

I Future Directions 94

I.1 Datasets . . . 94
I.1.1 Fusion . . . 94
I.1.2 Retrieval . . . 94
I.1.3 Question Answering . . . 94
I.1.4 Grounding . . . 94
I.1.5 Reinforcement Learning . . . 95

I.2 Models 95
I.3 Evaluation 95
I.3.1 Uncertainty Estimates 95
I.3.2 Robustness to Distribution Shifts 95
I.3.3 Fairness 96
I.4 Broader Outreach 97

A Broader Impact Statement

Multimodal data and models are ubiquitous in a range of real-world applications. MULTIBENCH and MULTIZOO is our aim to systematically categorize the plethora of datasets and models currently in use. While these contributions will accelerate research towards multimodal datasets and models as well as their real-world deployment, we believe that special care must be taken in the following regard to ensure that these models are safely deployed for real-world benefit:

Time & space complexity: Modern multimodal datasets and models are large, especially when building on already large pretrained unimodal datasets and models such as BERT or ResNets. The increasing time and space complexity of these models can cause financial impacts resulting from the cost of hardware, electricity, and computation, as well as environmental impacts resulting from the carbon footprint required to fuel modern hardware. Therefore, there has been much recent interest in building lightweight machine learning models [142].

MULTIBENCH also provides several efforts in this direction:

1. Firstly, MULTIBENCH alleviates the need for separate research groups to repeat preprocessing efforts when beginning to work on a new multimodal dataset, which often takes significant time when large video & audio datasets and feature extractors are involved.
2. Secondly, our standardized implementation of core approaches in MULTIZOO prevents duplicate efforts in adapting approaches to new datasets. We found that while many authors of these multimodal methods released their code publicly on GitHub, there was still some effort needed to adapt their code and tune their models to achieve the best performance on our standardized implementation in MULTIZOO. By standardizing these experimentation efforts, we can facilitate the sharing of code and trained models, ensure reproducibility across implementations, and save time and effort in the future.
3. Finally, MULTIBENCH explicitly tests for complexity and encourages researchers to build lightweight models. While this has been less studied in multimodal research, we hope that our efforts will pave the way for greener multimodal learning.

Privacy and security: There may be privacy risks associated with making predictions from multimodal data of recorded human behaviors. The datasets potentially in question might include those in affective computing (recorded video data labeled for sentiment, emotions, and personality attributes), and healthcare (health data labeled for disease and mortality rate). Therefore, it is crucial to obtain user consent before collecting device data. In our experiments with real-world data where people are involved (i.e., healthcare and affective computing), the creators of these datasets have taken the appropriate steps to only access public data which participants/content creators have consented for released to the public (see details in Appendix C.2). We only use these datasets for research purposes. All data was anonymized and stripped of all personal (e.g., personally identifiable information) and protected attributes (e.g., race, gender).

To deploy these algorithms at scale in the real world, it is also important to keep data and features private on each device without sending it to other locations using techniques such as federated learning [96, 100], differential privacy [55], or encryption [35].

Social biases: We acknowledge that there is a risk of exposure bias due to imbalanced datasets, especially when human-centric data and possibly sensitive labels are involved. For example, will models trained on imbalanced data disproportionately classify videos of a particular gender as displaying a particular emotion? Models trained on biased data have been shown to amplify the underlying social biases especially when they correlate with the prediction targets [108]. This leaves room for future work in exploring methods tailored for specific scenarios such as mitigating social biases in words [18], sentences [99], images [118], and other modalities. Future research in multimodal models should also focus on quantifying the trade-offs between fairness and performance [186]. MULTIBENCH enables the large-scale study of these crucial research questions and we outline some of our ongoing and future efforts in expanding the evaluation metrics in MULTIBENCH to take these into account in Appendix I.

Possible biases within each dataset: In this section, we expand upon the previous two points regarding privacy and social biases by describing the possible biases in each domain/dataset included in MULTIBENCH.

1. *Affective computing:* Analysis of sentiment, emotions, and personality might carry biases if care is not taken to appropriately anonymize the video data used. In MULTIBENCH, all models trained on affect recognition datasets use only pre-extracted non-invertible features that rely on general visual or audio features such as the presence of a smile or magnitude of voice. Therefore the features used in this paper cannot be used to identify the speaker [181, 183]. Furthermore, videos within the datasets all follow the creative commons license and follow fair use guidelines of YouTube. This license allows is the standard way for content creators to grant someone else permission to use and redistribute their work. We use no information regarding gender, ethnicity, identity, or video identifier in online sources. We emphasize that the models trained to perform automated affect recognition should not in any way be used to harm individuals and should only be used as a scientific study.
   In addition to privacy issues, we also studied the videos collected in these affective computing datasets and found no offensive content. While there are clearly expressions of highly negative sentiment or strong displays of anger and disgust, there are no offensive words used or personal attacks recorded in the video. All videos are related to movie or product reviews, TED talks, and TV shows.
2. *Healthcare:* The MIMIC dataset [78] has been rigorously de-identified in accordance with Health Insurance Portability and Accountability Act (HIPAA) such that all possible personal information has been removed from the dataset. Removed personal information include patient name, telephone number, address, and dates. Dates of birth for patients aged over 89 were shifted to obscure their true age. Please refer to Appendix C.2.2 for de-identification details. Again, we emphasize that any multimodal models trained to perform prediction should only be used for scientific study and should not in any way be used for real-world prediction.
3. *Finance:* There is no personal/human data included and there is no risk of personally identifiable information and offensive content.
4. *Robotics:* There is no personal/human data included and there is no risk of personally identifiable information and offensive content.
5. *HCI:* There is no personal/human data included and there is no risk of personally identifiable information and offensive content.
6. *Multimedia:* For MM-IMDb and AV-MNIST, there is no personal/human data included and there is no risk of personally identifiable information and offensive content. For Kinetics, all videos within the dataset are obtained from public YouTube videos that follow the creative commons license which allows content creators to grant permission to use and redistribute their work. We use no information regarding gender, ethnicity, identity, or video identifier in online sources. We emphasize that the models trained to perform action recognition should not in any way be used to harm individuals and should only be used as a scientific study.
   We also checked to make sure that these videos do not contain offensive content. All videos are related to human actions and do not contain any offensive words/audio.

Overall, MULTIBENCH offers opportunities to study these potential issues at scale across modalities, tasks, datasets, and domains. We plan to continue expanding this benchmark to rigorously test for these social impacts to improve the safety and reliability of multimodal models. For example, in Appendix I.3.3, we describe some concrete extensions to include evaluations for fairness and privacy of multimodal models trained on the datasets in MULTIBENCH. Our holistic evaluation metrics will also encourage the research community to quantify the tradeoffs between performance, complexity, robustness, fairness, and privacy in human-centric multimodal models.

B Background: Multimodal Representation Learning

We first provide background focusing on multimodal representation learning and several core technical challenges in this area.

B.1 Problem Statement

We define a modality as a single particular mode in which a signal is expressed or experienced. Multiple modalities then refer to a combination of multiple signals each expressed or experienced in heterogeneous manners [10]. We distinguish between the possible *temporal resolution* of modalities that will impact the types of approaches used:

1. *Static* modalities include inputs without a time dimension such as images, tabular data (i.e., a table of numerical data).

2. *Temporal* modalities include those coming in a sequence with a time-dimension such as language (a sequence of tokens), video (a sequence of frames/audio features/optical flow features), or time-series data (a sequence of data points indexed by time).

The first version of MULTIBENCH focuses on benchmarks and algorithms for *multimodal fusion*, where the main challenge is to join information from two or more modalities to perform a prediction. Classic examples include audio-visual speech recognition where visual lip motion is fused with speech signals to predict spoken words [48]. Note that in fusion problems, it should be well-defined to predict the label with a single modality only, which marks an important distinction to tasks in question answering and grounding where one modality is used to query information in another (e.g., visual question answering [4] using a text question to query information in the image). We outline our plans to extend future versions of MULTIBENCH to include more multimodal challenges such as question answering, retrieval, and grounding in Appendix I.

Formally, the multimodal fusion problem is defined as follows. We suppose there is a set of M modalities drawn from a joint distribution $p(X_1, ..., X_M, Y)$ where X_m is a random variable denoting data distributed according to modality m and Y is a random variable representing the label. If modality m is a static modality, X_m is a random vector without the time dimension. If modality m is a temporal modality, X_m is a random vector with a time dimension which can be represented as follows: $X_m = (X_m^1, X_m^2, ..., X_m^T)$ where T is the number of time-steps in the temporal modality.

In multimodal fusion, a set of M modalities is drawn from a joint distribution $p(X_1, ..., X_M, Y)$ where X_m is a random variable denoting data distributed according to modality m and Y is a random variable representing the label. A multimodal dataset is a collection of draws of (data, label) pairs from the joint distribution $p(X_1, ..., X_M, Y)$. We denote a dataset as $\{(\mathbf{x}_{i1}, ..., \mathbf{x}_{iM}, y_i)\}_{i=1}^n$. These draws from the true distribution are possibly biased (e.g., across individuals, topics, or labels) and noisy (e.g., due to noisy or missing modalities). A multimodal model is a set of functions $\{f_m : m \in [M]\} \cup f_{\text{mm}}$ where each of the f_m's are unimodal encoders, one for each modality, and f_{mm} is a multimodal fusion network. The unimodal encoders are specially designed with domain knowledge to learn representations from each modality (e.g., convolutional networks for images, temporal models for time-series data) resulting in unimodal representations $\mathbf{z}_1, ..., \mathbf{z}_M$. The multimodal network is designed to capture information across unimodal representations and summarize it in a multimodal representation $\mathbf{z}_{\text{mm}}$ that can be used to predict the label y. The goal of multimodal fusion is to learn a model with the lowest prediction error as measured on a held-out test set, while also balancing other potential objectives such as low complexity and robustness to imperfect data.

B.2 Technical Challenges

MULTIBENCH tests for the following holistic desiderata in multimodal fusion:

1. *Performance:* We summarize the following core challenges across all prediction tasks for multimodal learning with reference to Baltrusaitis et al., [10]. Solving these challenges is essential in any multimodal prediction problem, regardless of domain and task.
   (a) *Unimodal structure and granularity*: The information coming from each modality follows a certain underlying structure and invariance, which needs to be processed by suitable unimodal encoders. While there are certain generally adopted unimodal encoders for commonly studied modalities such as images and text, there remain challenges in designing unimodal encoders with the right types of inductive biases for other less-studied modalities such as tabular and time-series data. Representations

extracted from unimodal encoders should contain task-relevant information from that modality, expressed at the right granularity.

(b) *Multimodal complementarity*: The information coming from different modalities have varying predictive power by themselves and also when complemented by each other. We refer to these as *higher-order interactions*: first-order interactions define a predictive signal from a single granular unit of information in one modality to the label (e.g., the presence of a smile indicating positive sentiment); second-order interactions define a predictive signal from a pair of granular units of information across two modalities to the label (e.g., the presence of an eye-roll together with a positive word indicating sarcasm); and nth-order interactions extend the above definition to n modalities. There are many possible interactions that explain the labels in a dataset, out of which only some may generalize to unseen data. It remains a challenge to discover these higher-order interactions using suitably expressive models. At the same time, the space of possible interactions is too large which requires suitable inductive biases in model design (see challenges regarding complexity in model design below).

(c) *Multimodal alignment*: Information from different modalities often comes in different granularities. In order to learn predictive signals from higher-order interactions, there is a need to first identify the relations between granular units from two or more different modalities. This challenge requires a measure of the relationship between different modalities, which we call *cross-modal alignment*.

When dealing with temporal data, it also requires capturing possible long-range dependencies across time, which we call *temporal alignment*. For example, it requires aligning the presence of an eye-roll together with a positive word to recognize sarcasm even when both signals happen at different times. This challenge extends cross-modal alignment to the temporal dimension.

2. *Complexity:* The space of possible interactions is very large which requires suitable inductive biases in model design. While more expressive models may perform better, these often come at the cost of time and space complexity during training and inference. To enable real-world deployment of multimodal models in a variety of settings [142], there is a need to build lightweight models with cheap training and inference.
3. *Robustness:* Information from different modalities often display different noise topologies, and real-world multimodal signals possibly suffer from missing or noisy data in at least one of the modalities [10]. While most methods are trained on carefully curated and cleaned datasets, there is a need to benchmark their robustness in realistic scenarios. The core challenge here is to build models that still perform well despite the presence of unimodal-specific or multimodal imperfections.

Figure 5: MULTIBENCH provides a standardized machine learning pipeline across data processing, data loading, multimodal models, evaluation metrics, and a public leaderboard to encourage future research in multimodal representation learning. MULTIBENCH aims to present a milestone in unifying disjoint efforts in multimodal machine learning research and paves the way towards a better understanding of the capabilities and limitations of multimodal models, all the while ensuring ease of use, accessibility, and reproducibility.

C MULTIBENCH Datasets

MULTIBENCH provides a standardized machine learning pipeline that starts from data loading to running multimodal models, providing evaluation metrics, and a public leaderboard to encourage future research in multimodal representation learning (see Figure 5).

In this section, we provide additional details on the distribution, release, and maintenance of each of the datasets in MULTIBENCH as well as the maintenance of MULTIBENCH as a whole.

C.1 Dataset Selection

In this section, we discuss our choices of datasets in MULTIBENCH. We select each dataset based on its data collection method, input modalities, evaluation tasks, evaluation metric, and train/test splits that reflect real-world multimodal applications. We consulted with domain experts in each of the application areas to select datasets that satisfy the following properties:

1. *Realism in data collection, input modalities, preprocessing, and task:* Each of the datasets in MULTIBENCH reflect a subset of real-world sensory modalities collected in the wild. Realism is important since it brings natural noise topologies in each modality and in the prediction task. It is crucial that these datasets reflect real-world data such that capturing these imperfections through machine learning models can potentially bridge the gap towards real-world deployment.
2. *Diversity in research area:* We chose these research areas through a survey of recent research papers in multimodal learning across conferences in machine learning and beyond (e.g., HCI, NLP, vision, robotics conferences). Furthermore, we consulted with domain experts in applying multimodal learning to their respective application areas to determine areas of large potential. Through engaging with domain experts we were able to select research areas and datasets that reflected realism in data collection, input modalities, preprocessing, and tasks which present challenges for machine learning models and potential for real-world transfer of learned algorithms. These research areas that are designed to span both human-centric and data-centric machine learning. In the former, we selected HCI, healthcare, and robotics since these are fast-growing research areas with increasingly specialized tracks in machine learning conferences dedicated to them. In the latter, financial data analysis is an area with an inherently low signal-to-noise ratio reflecting extremely noisy, imperfect, and uncertain real-world datasets which provide challenges for current multimodal models. We also included several multimedia datasets due to the large resources publicly available on the internet which results in multimodal datasets of the largest scale.
3. *Diversity in modalities:* We started with a set of commonly studied modalities such as language, image, video, and audio. For each of the following research areas, we consulted with domain experts to choose datasets that are established, but not overstudied. More importantly, we aimed for diversity in modalities to truly test the generalization capabilities of modern multimodal models outside of commonly studied domains and modalities. For example, while there is much work in HCI involving images and text, we chose a modality representing a set of mobile applications for coverage. Similarly, in robotics, we consulted with domain experts to obtain datasets with high-frequency force and proprioception sensors that provide a unique challenge to machine learning researchers.
4. *Challenging for ML models:* We aim to choose datasets where the current state-of-the-art performance via machine learning models is still far from human performance (if human performance is provided, otherwise judged by a domain expert). This is to ensure that there is room for improvement through community involvement in this research area.

Figure 6: **Affective computing** studies the perception of human affective states (emotions, sentiment, and personalities) from our natural display of multimodal signals spanning language (spoken words), visual (facial expressions, gestures), and acoustic (prosody, speech tone) [124]. MULTIBENCH contains 4 datasets in this category involving fusing *language*, *video*, and *audio* time-series data to predict sentiment (CMU-MOSI [181] and CMU-MOSEI [183]), emotions (CMU-MOSEI [183]), humor (UR-FUNNY [64]), and sarcasm (MUSTARD [24]).

5. *Community expansion:* Finally, we would like to emphasize that we heavily encourage and actively seek out community participation in expanding MULTIBENCH to keep up with the incredible pace in multimodal machine learning research. We describe our plans for an open call for proposals of new research areas, datasets, and prediction tasks in section I.

C.2 Dataset Details

We provide details for each of the research areas and datasets selected in MULTIBENCH. In each of the categories, we describe the research area, the datasets and their associated data collection process, their access restrictions and licenses, and any data preprocessing or feature extraction we used following current work in each of these domains.

C.2.1 Affective Computing

1. MUSTARD is a multimodal video corpus for research in automated sarcasm discovery [24]. The dataset is compiled from popular TV shows including Friends, The Golden Girls, The Big Bang Theory, and Sarcasmaholics Anonymous. MUSTARD consists of audiovisual utterances annotated with sarcasm labels. Each utterance is accompanied by its context, which provides additional information on the scenario where the utterance occurs, thereby providing a further challenge in the long-range modeling of multimodal information. Sarcasm is specifically chosen as an annotation task since it requires careful modeling of complementary information, particularly when the semantic information from each modality *do not* agree with each other.

Data collection: According to Castro et al., [24], they conducted web searches on YouTube using keywords such as Friends sarcasm, Chandler sarcasm, Sarcasm 101, and Sarcasm in TV shows to obtain videos with sarcastic content from three main TV shows: Friends, The Golden Girls, and Sarcasmaholics Anonymous. To obtain non-sarcastic videos, they used a subset of 400 videos from MELD, a multimodal emotion recognition dataset derived from the Friends TV series [128]. Videos from The Big Bang Theory were also collected by segmenting episodes using laughter cues from its audience.

Access restrictions: While we do not have the license to this dataset, it is a public dataset free to download by the research community from `https://github.com/soujanyaporia/MUStARD`.

Licenses: MIT, see `https://github.com/soujanyaporia/MUStARD/blob/master/LICENSE`

Dataset preprocessing: We followed these preprocessing steps for each modality as suggested in the original paper [24]:

1. *Language:* Textual utterances are represented using pretrained BERT representations [42] as well as Common Crawl pre-trained 300-dimensional GloVe word vectors [119] for each token.
2. *Visual:* Visual features are extracted for each frame using a pool5 layer of an ImageNet [41] pretrained ResNet-152 [66] model. Every frame is first preprocessed by resizing, center-cropping, and normalizing it. We also use the OpenFace facial behavioral analysis tool [11] to extract facial expression features.

3. *Audio:* Low-level features from the audio data stream are extracted using the speech processing library Librosa [112]. We also extract COVAREP [39] features as is commonly used for the other datasets in the affective computing domain (see below).

Train, validation, and test splits: there are 414, 138, and 138 video segments in train, valid, and test data respectively, which gives a total of 690 data points.

2. CMU-MOSI is a collection of 2,199 opinion video clips each rigorously annotated with labels for subjectivity, sentiment intensity, per-frame, and per-opinion annotated visual features, and per-milliseconds annotated audio features [181]. Sentiment intensity is annotated in the range $[-3, +3]$ which enables fine-grained prediction of sentiment beyond the classical positive/negative split. Each video is collected from YouTube with a focus on video blogs, or vlogs which reflect the real-world distribution of speakers expressing their behaviors through monologue videos. CMU-MOSI is a realistic real-world multimodal dataset for affect recognition and is regularly used in competitions and workshops.

Data collection: According to Zadeh et al., [181], videos were collected from YouTube with a focus on video blogs indexed by #vlog. A total of 93 videos were randomly selected. The final set of videos contained 89 distinct speakers, including 41 female and 48 male speakers. Most of the speakers were approximately between the ages of 20 and 30 from different backgrounds (e.g., Caucasian, African-American, Hispanic, Asian). All speakers expressed themselves in English and the videos originated from either the United States of America or the United Kingdom.

Access restrictions: The authors are part of the team who collected the CMU-MOSI dataset [181] so we have the license and right to redistribute this dataset. CMU-MOSI was originally downloaded from https://github.com/A2Zadeh/CMU-MultimodalSDK.

Licenses: Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the conditions in https://raw.githubusercontent.com/A2Zadeh/CMU-MultimodalSDK/master/LICENSE.txt

Train, validation, and test splits: Each dataset contains several videos, and each video is further split into short segments (roughly 10 – 20 seconds) that are annotated. We split the data at the level of videos so that segments from the same video will not appear across train, valid, and test splits. This enables us to train user-independent models instead of having a model potentially memorizing the average affective state of a user. There are 52, 10, and 31 videos in train, valid, and test data respectively. Splitting up these videos gives a total of 1,284, 229, and 686 segments respectively for a total of 2,199 data points.

Dataset preprocessing: We follow current work [103, 183] and apply the following preliminary feature extraction for the CMU-MOSI dataset:

1. *Language:* Glove word embeddings [119] were used to embed a sequence of individual words from video segment transcripts into a sequence of word vectors that represent spoken text. The Glove word embeddings used are 300-dimensional word embedding trained on 840 billion tokens from the common crawl dataset, resulting in a sequence of dimension $T \times 300$ after alignment. The timing of word utterances is extracted using P2FA forced aligner [176]. This extraction enables alignment between text, audio, and video.
2. *Visual:* We use the library Facet [75] to extract a set of visual features including facial action units, facial landmarks, head pose, gaze tracking, and HOG features [189]. These visual features are extracted from the full video segment at 30Hz to form a sequence of facial gesture changes throughout time, resulting in a sequence of dimension $T \times 35$. In addition to Facet, OpenFace facial behavioral analysis tool [11] is used to extract the facial expression features which include facial Action Units (AU) based on the Facial Action Coding System (FACS) [49].
3. *Audio:* The software COVAREP [39] is used to extract acoustic features including 12 Mel-frequency cepstral coefficients, pitch tracking and voiced/unvoiced segment features [46], glottal source parameters [28], peak slope parameters and maxima dispersion quotients [79]. These visual features are extracted from the full audio clip of each segment at 100Hz to

form a sequence that represents variations in tone of voice over an audio segment, resulting in a sequence of dimension $T \times 74$.

3. UR-FUNNY is the first large-scale multimodal dataset of humor detection in human speech [64]. UR-FUNNY is a realistic representation of multimodal language (including text, visual and acoustic modalities). This dataset opens the door to understanding and modeling humor in a multimodal framework, which is crucial since humor is an inherently multimodal communicative tool involving the effective use of words (text), accompanying gestures (visual), and prosodic cues (acoustic). UR-FUNNY consists of more than $16,000$ video samples from TED talks which are among the most diverse idea-sharing channels covering speakers from various backgrounds, ethnic groups, and cultures discussing a variety of topics from discoveries in science and arts to motivational speeches and everyday events. The diversity of speakers, topics, and unique annotation targets make it a realistic dataset for multimodal language modeling.

Data collection: According to Hasan et al., [64] $1,866$ videos and their transcripts in English were collected from the TED portal, chosen from $1,741$ different speakers and across 417 topics. The laughter markup is used to filter out 8257 humorous punchlines from the transcripts. The context is extracted from the prior sentences to the punchline (until the previous humor instances or the beginning of the video is reached). Using a similar approach, $8,257$ negative samples are chosen at random intervals where the last sentence is not immediately followed by a laughter marker. After this negative sampling, there is a homogeneous 50% split in the dataset between positive and negative humor examples.

Access restrictions: This is a public dataset free to download by the research community from https://github.com/ROC-HCI/UR-FUNNY. The authors of the dataset also note that videos on www.ted.com are publicly available for download [64].

Licenses: No license was provided with this dataset.

Dataset preprocessing: We follow current work [103, 183] and apply the same preliminary feature extraction as the CMU-MOSI dataset described above.

Train, validation, and test splits: Each dataset contains several videos, and each video is further split into short segments (roughly $10-20$ seconds) that are annotated. We split the data at the level of videos so that segments from the same video will not appear across train, valid, and test splits. This enables us to train user-independent models instead of having a model potentially memorizing the average affective state of a user. There are $1,166$, 300, and 400 videos in train, valid, and test data respectively. Splitting up these videos gives a total of $10,598$, $2,626$, and $3,290$ segments respectively for a total of $16,514$ data points,

4. CMU-MOSEI is the largest dataset of sentence-level sentiment analysis and emotion recognition in real-world online videos [102, 183]. CMU-MOSEI contains more than 65 hours of annotated video from more than $1,000$ speakers and 250 topics. Each video is annotated for sentiment as well as the presence of 9 discrete emotions (angry, excited, fear, sad, surprised, frustrated, happy, disappointed, and neutral) as well as continuous emotions (valence, arousal, and dominance). The diversity of prediction tasks makes CMU-MOSEI a valuable dataset to test multimodal models across a range of real-world affective computing tasks. The dataset has been continuously used in workshops and competitions revolving around human multimodal language.

Data collection: According to Zadeh et al., [183], videos from YouTube are automatically analyzed for the presence of one speaker in the frame using face detection to ensure the video is a monologue and rejecting videos that have moving cameras. A diverse set of 250 frequently used topics in online videos is used as the seed for acquisition. The authors restrict the number of videos acquired from each channel to a maximum of 10 and limit the videos to have manual and properly punctuated transcriptions. After manual quality inspection, they also performed automatic checks on the quality of video and transcript using facial feature extraction confidence and forced alignment confidence, before balancing the gender in the dataset using the data provided by annotators (57% male to 43% female).

Access restrictions: The authors are part of the team who collected the CMU-MOSEI dataset [183] so we have the license and right to redistribute this dataset. CMU-MOSEI was originally downloaded from https://github.com/A2Zadeh/CMU-MultimodalSDK.

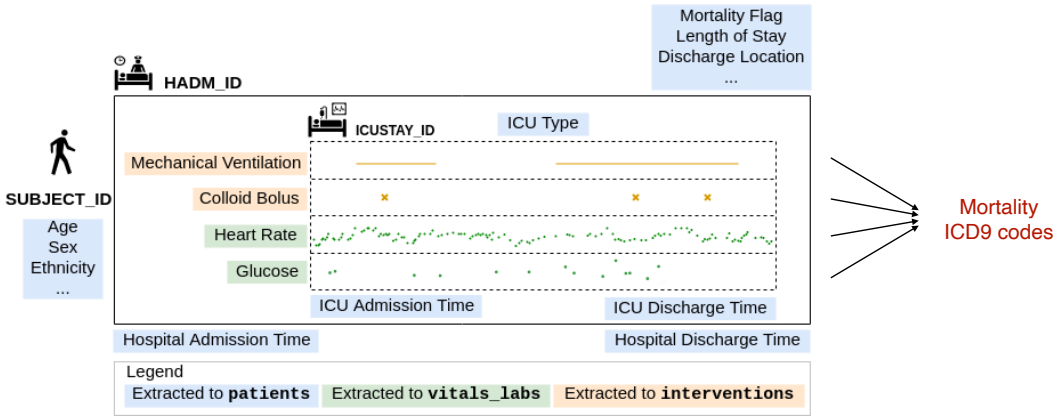


Figure 7: **Healthcare:** Medical decision-making often involves integrating complementary signals from several sources such as lab tests, imaging reports, and patient-doctor conversations. Multimodal models can help doctors make sense of high-dimensional data and assist them in the diagnosis process [5]. MULTIBENCH includes the MIMIC dataset [78] which records ICU patient data including *time-series* data measured every hour and other *tabular numerical* data about the patient (e.g., age, gender, ethnicity) to predict mortality rate and the disease ICD-9-code. Figure adapted from [165].

Licenses: Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the“"Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the conditions in https://raw.githubusercontent.com/A2Zadeh/CMU-MultimodalSDK/master/LICENSE.txt

Dataset preprocessing: We follow current work [103, 183] and apply the same preliminary feature extraction as the CMU-MOSI and UR-FUNNY datasets described above.

Train, validation, and test splits: Each dataset contains several videos, and each video is further split into short segments (roughly $10-20$ seconds) that are annotated. We split the data at the level of videos so that segments from the same video will not appear across train, valid, and test splits. This enables us to train user-independent models instead of having a model potentially memorizing the average affective state of a user. There are a total of $16,265$, $1,869$, and $4,643$ segments in train, valid, and test datasets respectively for a total of $22,777$ data points.

C.2.2 Healthcare

1. MIMIC-III (Medical Information Mart for Intensive Care III) [78] is a large, freely-available database comprising de-identified health-related data associated with over $40,000$ patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. Following [129], we organized numerous patient data into two major modalities (using the 17 features in feature set A in [129]): time series modality, which is a set of medical measurements of the patient taken every 1 hour in a period of 24 hours where each measurement is a vector of size 12 (12 different measured numerical values); static modality, which is a set of medical information about the patient, represented in a vector of size 5. We use these modalities for 3 tasks: mortality prediction (6-class prediction on whether the patient dies in 1 day, 2 day, 3 day, 1 week, 1 year, or longer than 1 year), and 2 ICD-9 code predictions (binary classification on whether the patient fits any ICD-9 code in group 1 ($140-239$) and binary classification on whether the patient fits any ICD-9 code in group 7 $460-519$).

Data collection: According to Johnson et al., [78], MIMIC contains data associated with $53,423$ distinct hospital admissions for adult patients (aged 16 years or above) admitted to critical care units between 2001 and 2012, as well as $7,870$ neonates admitted between 2001 and 2008. The data covers $38,597$ distinct adult patients and $49,785$ hospital admissions. Data was also downloaded from several sources, including archives from critical care information systems, hospital electronic health record databases, and Social Security Administration Death Master File.

Privacy: Before data was incorporated into the MIMIC-III database, it was first de-identified in accordance with Health Insurance Portability and Accountability Act (HIPAA) standards using structured data cleansing and date shifting. The de-identification process removed all eighteen identifying data elements listed in HIPAA, such as patient name, date of birth (for patients over 89 of age), telephone number, address, and dates. Protected health information was also removed from text fields, such as diagnostic reports and physician notes. We refer the reader to [129] for full de-identification details.

Access restrictions: We do not have the license and right to redistribute this dataset. Accessing MIMIC requires the completion of a training course and approval for access on PhysioNet (https://physionet.org/about/database/). However, we provide our own data preprocessing scripts for MIMIC, which transform the raw data into the standardized format for multimodal data and perform standardized splitting into the train, validation, and test splits. For a new user getting started with MIMIC data, all they would need to do is to complete the training course and obtain approval of access for scientific research from PhysioNet before they can use our public code to load all extracted features from the raw dataset in a version that can directly be used for machine learning studies.

Licenses: MIT, see https://github.com/mit-lcp/mimic-code/blob/main/LICENSE

Dataset preprocessing: We followed the instructions on https://mimic.physionet.org/gettingstarted/access/ to download the dataset in the form of raw tables, then generated preprocessed data following the steps described in https://github.com/USC-Melady/Benchmarking_DL_MIMICIII (which takes 1 – 2 weeks running time) to get the data used for experiments. Specifically, we will use data in the file `24hrs/series/imputed-normed-ep_1_24-stdized.npz`. When accessing this data from our code repo, set the `imputed_path` of the npz file above in the `get_data.py` and the script will generate the PyTorch data loader for the tasks (where we will normalize the data).

Train, validation, and test splits: We split the data into train/valid/test sets randomly (using a fixed random seed) in a $80:10:10$ ratio (so $28,970$ train, $3,621$ valid, and $3,621$ test data points) for a total of $36,212$ data points.

C.2.3 Robotics

1. MuJoCo Push is a planar pushing task, in which a 7-DoF Panda Franka robot is pushing a circular puck with its end-effector in simulation. We estimate the 2D position of the unknown object on a table surface, while the robot intermittently interacts with the object. Similar to Vision&Touch, planar pushing is a contact-rich task. However, instead of estimating robot states, this dataset is estimating the state of the object the robot is currently interacting with. While other robotics datasets have also studied planar pushing [14, 175], Yu et al., [175] use a Vicon tracker (instead of raw RGB images) while Bauza et al., [14] only collect visual and proprioceptive data.

Data collection: According to Lee et al. [90], this dataset consists of 1000 trajectories with 250 steps at 1.0×10^1 Hz, of a simulated Franka Panda robot arm pushing a circular puck in MuJoCo [152]. The pushing actions are generated by a heuristic controller that tries to move the end-effector to the center of the object. The multimodal inputs are gray-scaled images $(1 \times 32 \times 32)$ from an RGB camera, forces (and binary contact information) from a force/torque sensor, and the 3D position of the robot end-effector. The task is to predict the 2-D planar object pose which we measure by MSE.

Access restrictions: While we do not have the license to this dataset, it is a public dataset free to download by the research community from https://github.com/brentyi/multimodalfilter/.

Licenses: MIT, see https://github.com/brentyi/multimodalfilter/blob/master/LICENSE.

Dataset preprocessing: Training, validation, and test data are each in their own files and can be used directly after downloading. Data is normalized using mean and variance from the train set.

Train, validation, and test splits: This dataset contains 1000 training data, 10 validation data, and 300 test data. Each data point is split into 29 time-series sequences of length 16. The total number

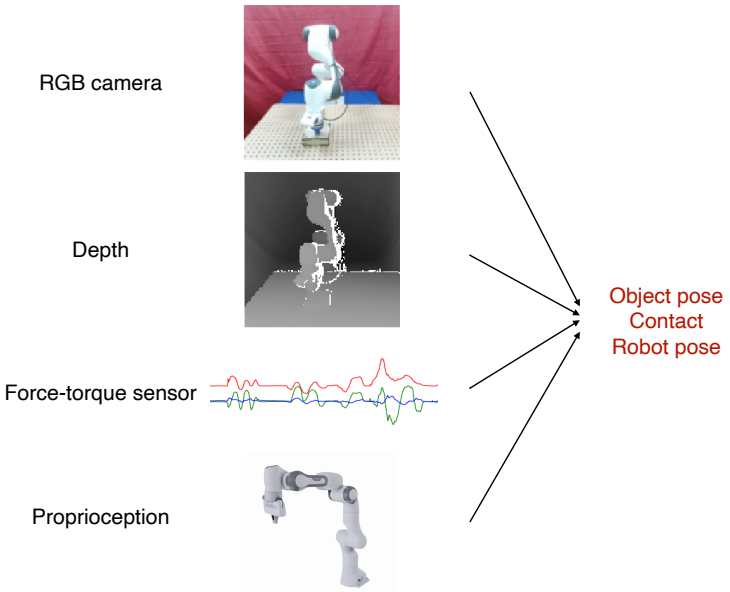


Figure 8: **Robotics:** Modern robot systems are equipped with multiple sensors to aid in their decision-making. We include the large-scale MuJoCo Push [90] and Vision&Touch [92] datasets which record the manipulation of real and simulated robotic arms equipped with visual (RGB and depth), force, and proprioception sensors. In MuJoCo Push, the goal is to predict the pose of the object being pushed by the robot end-effector. In Vision&Touch, the goal is to predict action-conditional learning objectives that capture forward dynamics of the different modalities (contact prediction and robot end-effector pose). Figure adapted from [91].

of data points for training, validation, and test are $29,000$, 290, and $8,700$ for a total of 37990 data points.

2. Vision&Touch is a real-world robot manipulation dataset that collects visual, force, and robot proprioception data (as well as the robot actions) for a peg insertion task. The robot is a 7-DoF, torque-controlled Franka Panda robot, which has a triangle peg attached to its end-effector. Rigidly attached to the table in front of the robot is a box with a triangle hole. The robot attempts to insert the peg into the hole, a contact-rich manipulation task that has been studied for decades due to its relevance in manufacturing. Vision, force, and proprioception are feedback modalities shown to be complementary and concurrent during contact-rich manipulation [17].

Data collection: According to Lee et al., [92], the data is collected by running on the robot a random policy (that takes random actions) as well as a heuristic policy (that attempts peg insertion). Four sensor modalities are available, including robot proprioception, an RGB-D camera, and a force-torque sensor. The proprioceptive input is the robot end-effector pose as well as linear and angular velocity. They are computed using forward kinematics. RGB images and depth maps are recorded from a fixed camera (Kinect v2 camera) pointed at the robot. Input images to our model are down-sampled to 128×128. The force sensor provides 6-axis feedback on the forces and moments along the x, y, z axes. The OptoForce force sensor is mounted between the last joint and the peg. The robot action data is also collected at every timestep. The robot action is the Cartesian end-effector position displacement and z-axis roll rotation of the end-effector. There are 150 trajectories collected, each with 1000 timesteps of data collected. While the dataset originally was intended for representation learning for reinforcement learning, We use 2 tasks from the Vision&Touch datasets: (1) predicting binary contact in the next time step and (2) predicting end-effector position measured in MSE.

Access restrictions: While we do not have the license to this dataset, it is a public dataset free to download by the research community from https://github.com/stanford-iprl-lab/multimodal_representation/.

Licenses: MIT, see https://github.com/stanford-iprl-lab/multimodal_representation/blob/master/LICENSE.

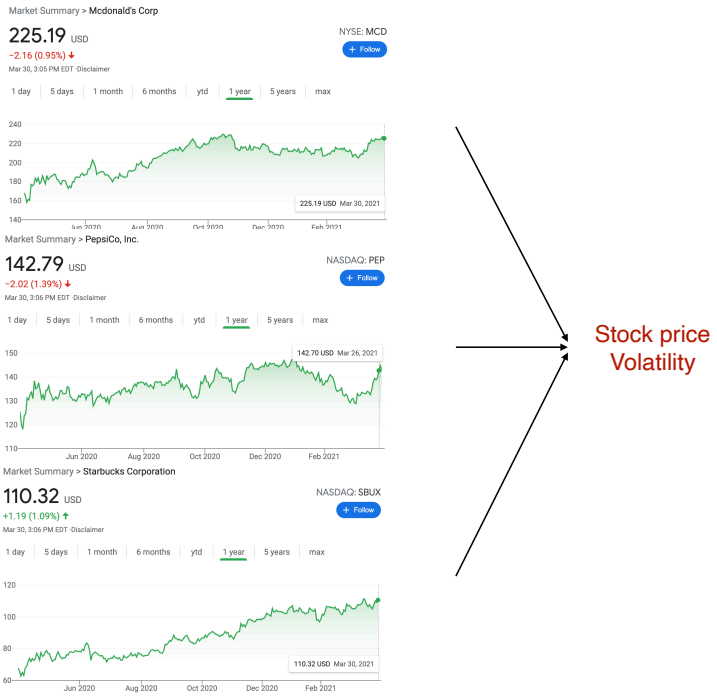


Figure 9: **Finance:** We scrape historical stock data from the internet and create our own dataset for financial time-series prediction across 3 groups of correlated stocks: STOCKS-F&B, STOCKS-HEALTH, and STOCKS-TECH. Within each group, the previous stock prices of a set of stocks are used as multimodal input to predict the squared return of a related stock (e.g., using Apple, Google, and Microsoft historical data to predict future prices of Microsoft).

Dataset preprocessing: Dataset has already been pre-processed and can be downloaded directly at https://github.com/stanford-iprl-lab/multimodal_representation/. The dataset comes as a zipped file with 3000 hdf5 files, each with 50 timesteps of data. In order to get action-conditional contact as well as robot end-effector position, the dataset uses the contact and end-effector position data from the next timestep. Since the data from the first time step cannot be used, only 49 of 50 timesteps of data per file can be used.

Train/validation split: This dataset uses a $80:20$ training and validation split. There are 117600 training data points and 29400 validation data points. Since the original dataset does not contain test data, we report validation performance instead of test performance for this dataset.

C.2.4 Finance

We created the following financial datasets which consist of historical stock data retrieved from publicly available online financial databases. We record the opening price of each stock from 2000-06-01 to 2021-02-28, which creates a total of 5218 time steps. Details of each dataset are described in its own section below.

1. STOCKS-F&B consists of 18 selected stocks from S&P 500 stocks categorized by GICS as Restaurants or Packaged Foods & Meats. We select MCD, SBUX, HSY, and HRL for initial experiments on this dataset, record their opening prices, and preprocess the data following the preprocessing procedures below.

2. STOCKS-HEALTH consists of 63 selected stocks from S&P 500 stocks categorized by GICS as Health Care. We select MRK, WST, CVS, MCK, ABT, UNH, and TFX for initial experiments on this dataset, record their opening prices, and preprocess the data following the preprocessing procedures below.

3. STOCKS-TECH consists of 100 selected stocks from S&P 500 stocks categorized by GICS as Information Technology or Communication Services. We select AAPL, MSFT, AMZN, INTC, AMD,

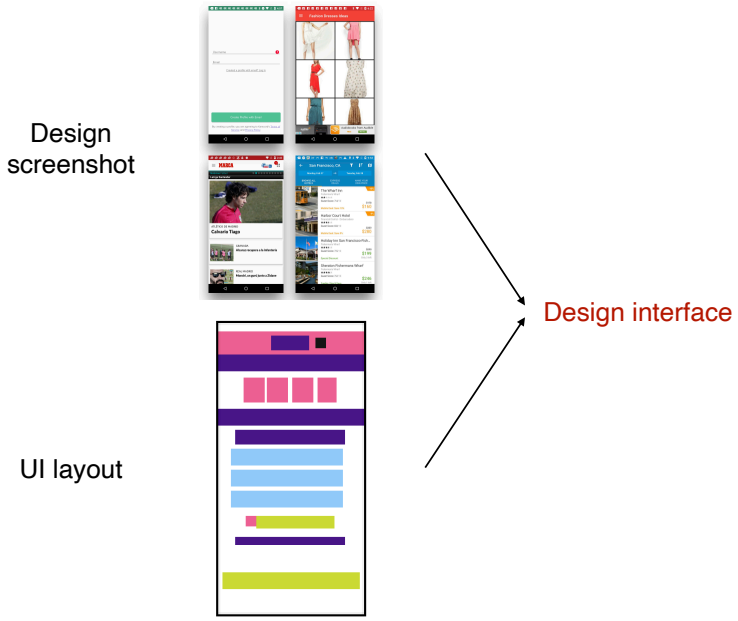


Figure 10: **Human Computer Interaction (HCI)** studies the design and use of computer technology with a focus on the interactive interfaces between humans and computers. We use the ENRICO (Enhanced Rico) dataset [40, 93] of Android app screens (consisting of an image as well as a set of apps and their locations) categorized by their design motifs and collected for data-driven design applications such as design search, user interface (UI) layout generation, UI code generation, and user interaction modeling. Figure adapted from [40, 93].

and `MSI` for initial experiments on this dataset, record their opening prices, and preprocess the data following the preprocessing procedures below.

Access restrictions: The datasets were collected from Yahoo Finance, which is publicly available but does not allow redistribution of their data. We provide automated download and preprocessing scripts for this dataset.

Licenses: We could not find a finance dataset with a free redistribution license that includes historical financial data. As such, we provide automated download and preprocessing scripts as part of this project, which utilizes the open-source `pandas-datareader` to download raw finance data. We used the open-source code at `https://github.com/pydata/pandas-datareader/blob/master/pandas_datareader/yahoo/components.py`. The automated scripts we provide are licensed under an MIT License.

Dataset preprocessing: Data is downloaded, converted to returns, and normalized. Labels are converted to squared returns. Each time series is split in chronological order, where the test split corresponds to the latest prices. For each data point, 500 previous returns are used to predict the squared return of the next day. The first 500 time steps are not predicted since they do not have 500 previous steps. We consider each stock as a modality; unimodal datasets have the input stock identical to the target stock. To keep memory usage practical for MULT [154] models, we evenly separate the stocks into 3 groups and use each group as a modality when preprocessing for MULT [154].

Train, validation, and test splits: We split the data according to time. There are 3200 continuous days of stock prices in the train data (2002-06-04 start to 2015-02-18 end date), 500 continuous days of stock prices in the valid data (2015-02-19 start to 2017-02-10 end date), and 1017 continuous days of stock prices in the test data (2017-02-13 start to 2021-02-26 end date).

C.2.5 HCI

1. ENRICO (Enhanced Rico) [93] is a dataset of Android app screens categorized by their design motifs. ENRICO was collected to help data-driven design applications such as design search, UI layout generation, UI code generation, and user interaction modeling. ENRICO is a subset of RICO [40],

which is a large dataset of app screens collected by the automated and semi-automated "crawling" of Android apps available on the Google Play Store.

The RICO and ENRICO datasets have been used as benchmarks for data-driven models of design in scaffolding the creation of mobile apps. These constitute a set of relevant examples that help designers understand best practices and trends in building human-centered interfaces. Building multimodal models on these examples will enable systems that can predict whether a UI design will achieve its targeted goals even before it is deployed to millions of people. In the long run, this will enable the large-scale creation of personalized UI designs that can automatically adapt to diverse users and contexts.

The authors of ENRICO employed two main modalities for app classification: (1) the app screenshot and (2) the view hierarchy. The app screenshot is given in the form of an image. The view hierarchy is a type of metadata associated with some UI screens that describe the spatial and structural layout of UI elements. This view hierarchy can be treated as a *set* since it contains an unordered collection of UI elements each containing metadata and their spatial and structural layout.

Data collection: The original RICO dataset was collected using a combination of manual (*i.e.,* crowdworkers) and automated (*i.e.,* app crawler) methods. More information about how the apps were downloaded and captured is available in the RICO paper [40]. The ENRICO dataset is a subset of RICO that was created by first randomly sampling 10000 screens from RICO and labeling a high-quality subset (1460 screens) that can be categorized into 20 design categories. More information about the collection and annotation process is available in the ENRICO paper [93].

Access restrictions: While we do not have the license to this dataset, it is a public dataset free to download by the research community from https://github.com/luileito/enrico.

Licenses: MIT, see https://github.com/luileito/enrico/blob/master/LICENSE

Dataset preprocessing: We extract the following features from each modality:

1. *Image:* The authors of ENRICO used a VGG-16 network (augmented with batch normalization and dropout) to encode app screenshots. To reduce overfitting on the relatively small dataset (1460 examples), we use a VGG-11 network pre-trained on ImageNet, with a frozen feature extraction network and a slimmed-down classifier network.
2. *Set:* We followed prior modeling approaches [40, 93] to represent the view hierarchy as a set of UI elements spatially rendered as a "wireframe" (similar to a semantic map). The wireframe was then fed into the same VGG-11 network used to encode the screenshot. Another possibility, which we briefly explored, is to use a set encoder [184] to use a permutation invariant function to compute a pooled representation of the set of mobile applications. We found that the CNN-based approach resulted in better performance, as it allowed the network to be initialized from a pre-trained checkpoint, although our experiments were initial and there is still ample room for future work to explore better encoders for this set modality.

Train, validation, and test splits: The original paper doesn't provide official splits for training, validation, and testing. We used a known seed to deterministically shuffle the dataset and create splits for training (65%, 947 examples), validation (15%, 219 examples), and testing (20%, 292 examples).

C.2.6 Multimedia

1. AV-MNIST is a multimodal dataset created by paring audio of a human reading digits from the FSDD dataset [1] with written digits in the MNIST dataset [88] with a task to predict the digit into one of 10 classes (0 – 9). Since existing models can already complete the digit recognition task from either modality quite well, one common practice in previous work [161] is to increase the difficulty by removing 75% of energy in the visual modality via PCA and adding noise from ESC-50 [125] to the audio modality, such that models have to leverage information from both modalities to make accurate predictions. ESC-50 is a realistic dataset collected from real-world audio of various everyday objects. Therefore, AV-MNIST serves as a good starting point of a relatively simple multimodal dataset but with underlying challenges of complementarity and noisy data. In fact, the method of injecting real-world background noises into the audio modality also inspired more tests for robustness included in MULTIBENCH. AV-MNIST has served as a popular benchmark for evaluating the effectiveness of multimodal fusion models [122, 161].

Data collection: According to Vielzeuf et al., [161], AV-MNIST starts with the entirety of the MNIST image and FSDD audio datasets. The audio samples are augmented by adding randomly chosen 'noise' samples from the ESC-50 dataset [125], to reach the same number of examples as in MNIST (55000 training, 5000 validation, and 10000 testing examples).

Access restrictions: This dataset is programmatically generated by combining 2 unimodal datasets: MNIST and FSDD (with the additional audio signal from ESC-50). While we do not have the license to these datasets, they are public datasets free to download by the research community.

Licenses: MNIST is released with a Creative Commons Attribution-Share Alike 3.0. FSDD is released with a Creative Commons Attribution-ShareAlike 4.0 International license. ESC-50 is released with a Creative Commons Attribution Non-Commercial license. All of these licenses allow redistribution of the datasets.

Dataset preprocessing: To create the dataset, we downloaded MNIST from http://yann.lecun.com/exdb/mnist/, FSDD from https://github.com/Jakobovski/free-spoken-digit-dataset, ESC-50 from https://github.com/karolpiczak/ESC-50, and generated AV-MNIST with the scripts provided in https://github.com/slyviacassell/_MFAS/blob/master/datasets/avmnist_gen.py. Note that since the official implementation of the preprocessing is not released, our preprocessing, as well as all other existing preprocessing scripts, may differ from the original preprocessing in some details (such as keeping at most or at least 25% of energy in the image modality, and some parameters in adding noise to audio), so the performance of models in our version of AV-MNIST should not be compared directly with the performance of models on AV-MNIST in other papers.

No preprocessing is done for the image modality. For audio, it is converted to a $112x112$ Spectogram. See the code in https://github.com/slyviacassell/_MFAS/blob/master/datasets/avmnist_gen.py for details.

Train, validation, and test splits: Data splits for AV-MNIST follow that of the MNIST dataset, with 55000 training, 5000 validation, and 10000 testing examples.

2. MM-IMDB is the largest publicly available multimodal dataset for genre prediction on movies [8]. MM-IMDB starts from the movies of the MovieLens 20M dataset and expands this dataset by collecting genre, poster, and plot information for each movie. The final dataset contains ratings for $25,959$ movies. MM-IMDB is a realistic real-world multimodal dataset and is a popular benchmark for multimodal learning [8, 81, 122].

Data collection: According to Arevalo et al., [8], MM-IMDB dataset is built with the IMDb ids provided by the Movielens 20M dataset that contains ratings of $27,000$ movies. Using the IMDbPY 3 library, movies that do not contain their poster image were filtered out. The resulting dataset comprises $25,959$ movies along with their plot, poster, genres, and other 50 additional metadata fields such as year, language, writer, director, aspect ratio, etc. The task is to perform multilabel classification into one of 23 movie genres.

Access restrictions: While we do not have the license to this dataset, it is a public dataset free to download by the research community from http://lisi1.unal.edu.co/mmimdb/ and https://github.com/johnarevalo/gmu-mmimdb/.

Licenses: MIT, see https://github.com/johnarevalo/gmu-mmimdb/blob/master/LICENSE

Dataset preprocessing: We used the same method as [8] to extract features from texts and images.

1. *Text:* We used the pretrained Google Word2vec[1] to extract text features. The final vocabulary contains $41,612$, which is the intersection of Google word2vec words and the MM-IMDB plots. We converted all text to lowercase following existing work.
2. *Image:* All images were scaled, and cropped when required, to 160×256 pixels keeping the aspect ratio. A VGG-16 model [139] is applied as the image feature extractor. This CNN consists of 5 convolutional layers of $5, 3, 3, 3, 3$ squared filters and 2×2 pool sizes. Each convolutional layer has 16 hidden units. The convolutional layers are connected with a MaxoutMLP on top.

[1] https://code.google.com/archive/p/word2vec/

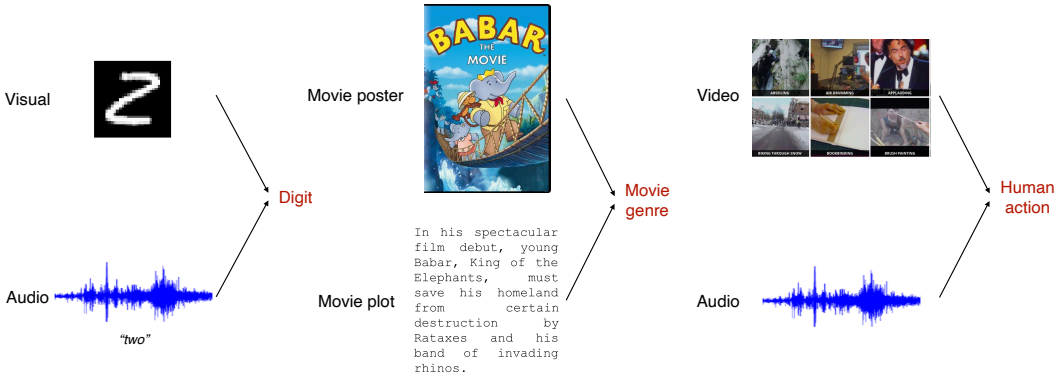


Figure 11: **Multimedia:** A significant body of research in multimodal learning has been fueled by the large availability of multimedia data (language, image, video, and audio) on the internet. MULTIBENCH includes 3 popular large-scale multimedia datasets with varying sizes and levels of difficulty: (1) Audio-Visual MNIST (AV-MNIST) [161] is assembled from images of handwritten digits [88] and audio samples of spoken digits [94], (2) Multimodal IMDb (MM-IMDB) [8] uses movie titles, metadata, and movie posters to perform multi-label classification to a set of movie genres, and (3) KINETICS [80] contains video and audio of $306,245$ video clips annotated for 400 human actions. To ease experimentation, we split KINETICS into small and large partitions (see Appendix C). Figure adapted from [8, 80].

Train, validation, and test splits: The MM-IMDb dataset is split by genre into train, valid, and test datasets containing 15552, 2608, and 7799. The split was performed so that training, valid and test sets comprise 60%, 10%, 30% samples of each genre respectively.

3. KINETICS is a series of large-scale curated video clips covering a diverse range of human actions. We use the original Kinetics-400 dataset [80] which contains 400 human action classes, with at least 400 video clips for each action. Each clip lasts around 10s and is taken from a different YouTube video. This is one of the largest publicly available multimodal datasets with a total of $306,245$ video clips spanning 400 human actions. Therefore, KINETICS is suitable for testing the scalability of multimodal models to extremely large datasets. Furthermore, recognizing human actions is a core challenge in a variety of applications such as human-AI interaction, robotics, and human behavior analysis.

The sheer scale of the KINETICS dataset means that even the simplest models take up to several weeks to finish training. To enable multimodal learning from video and audio while also increasing access across researchers with limited computing resources, we subsample the KINETICS dataset into small and large partitions:

KINETICS-S: We subsampled 5 human actions: *archery*, *breakdancing*, *crying*, *dining*, *singing* and retained all video clips annotated for these 5 actions. We selected these actions randomly out of the 400 actions in Kinetics-400. This gave us a total of 2624 video clips in the small version of the dataset. Training a basic supervised learning model on KINETICS-S takes roughly 2 hours on a single GPU.

KINETICS-L: This represents the entire KINETICS-400 dataset with $306,245$ video clips spanning 400 human actions. Training a basic supervised learning model on KINETICS-L takes roughly 2 weeks on a single GPU.

Data collection: We refer the reader to Kay et al., [80] for a detailed description of the dataset collection process. Briefly, the authors (1) started with a list of human actions from sources spanning existing action datasets, motion capture, and crowdsourcing, (2) obtained candidate clips from YouTube and extracted temporal positions within a video, (3) performed manual labeling for human actions with Amazon's Mechanical Turk, and (4) cleaning up and de-noising the selected videos.

Access restrictions: While we do not have the license to this dataset, it is a public dataset free to download by the research community from https://deepmind.com/research/open-source/kinetics.

Licenses: Creative Commons Attribution 4.0 International, so we are free to share, copy, and redistribute the material in any medium or format, see https://deepmind.com/research/open-source/kinetics.

Dataset preprocessing: We downloaded links from https://deepmind.com/research/open-source/kinetics and preprocessed them with the torchvision Kinetics scripts.

We processed the video and audio modalities as follows:

1. *Video:* We use $150 \times 224 \times 224 \times 3$ input clips, created with a frame skip of 2, a center crop with shape $(224, 224)$, and the normalization step required for using torchvision.models.
2. *Audio:* We use log-scaled mel spectrograms with 763 temporal frames by 40 Mel filters, element-wise averaging 2-channel waveforms to yield single channel ones.

Train, validation, and test splits: We use the $80.5/6.5/13$ split provided by the original dataset, taking all the data points in our chosen classes. This yields 2112, 171, and 341 data points in train, validation, and test splits respectively for KINETICS-S and 246527, 19906, and 39812 data points in train, validation, and test splits respectively for KINETICS-L.

C.3 Documentation

We provide documentation for MULTIBENCH in the form of datasheets for datasets [54]:

1. **Motivation**
   (a) *For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.*
   Learning multimodal representations involves integrating information from multiple heterogeneous sources of data. It is a challenging yet crucial area with numerous real-world applications in multimedia, affective computing, robotics, finance, and healthcare. Unfortunately, current research focuses primarily on a fixed set of modalities and tasks without a concrete understanding of generalization across domains and modalities, complexity during training and inference, and robustness to noisy and missing modalities. In order to standardize multimodal research and accelerate progress towards understudied modalities and tasks while ensuring real-world robustness, we release MULTIBENCH, a systematic and unified large-scale benchmark for multimodal learning spanning 15 datasets, 10 modalities, 20 prediction tasks, and 6 research areas. MULTIBENCH provides an automated end-to-end machine learning pipeline that simplifies and standardizes data loading, experimental setup, and model evaluation. To enable holistic evaluation, MULTIBENCH summarizes both performance as well as the potential drawbacks involving increased time and space complexity and risk of decreased robustness from other modalities. To accompany MULTIBENCH, we also provide a standardized implementation of 20 core approaches in multimodal learning unifying innovations in fusion paradigms, optimization objectives, and training approaches.
   MULTIBENCH datasets present significant challenges of scalability to large-scale multimodal datasets and robustness to realistic imperfections, which present fruitful opportunities for future research. We hope that MULTIBENCH will present a milestone in unifying disjoint efforts in multimodal machine learning research and paves a way towards a better understanding of the capabilities and limitations of multimodal models, all the while ensuring ease of use, accessibility, and reproducibility. MULTIBENCH, our standardized implementation, and leaderboards are publicly available, will be regularly updated, and welcomes inputs from the community.
   (b) *Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?*
   MULTIBENCH is created primarily by the MultiComp Lab in the Language Technologies Institute and Machine Learning Department of the School of Computer Science at Carnegie Mellon University, in collaboration with several other researchers in the Human-Computer Interaction Institute and Computer Science Department at Carnegie Mellon University as well as at Johns Hopkins University, Stanford University, and UT Austin. The creation of MULTIBENCH is for purely research purposes only.
   (c) *Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.*

This material was based upon work partially supported by the National Science Foundation (Awards #1722822 and #1750439) and National Institutes of Health (Awards #R01MH125740, #R01MH096951, #U01MH116925, and #U01MH116923), NSF IIS1763562, and ONR Grant N000141812861. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or National Institutes of Health, and no official endorsement should be inferred.

(d) *Any other comments?*
No.

2. **Composition**

(a) *What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.*
We describe each dataset in detail in Appendix C.2. MULTIBENCH provides a comprehensive suite of multimodal datasets to benchmark current and proposed approaches in multimodal representation learning. It covers a diverse range of research areas (affective computing, healthcare, robotics, finance, HCI, and multimedia), dataset sizes (small, medium, and large), input modalities (in the form of ℓ: language, i: image, v: video, a: audio, t: time-series, ta: tabular, o: optical flow, f: force sensor, p: proprioception sensor, s: set), and prediction tasks (affect recognition, robot manipulation, stock prediction, design interface, action recognition, movie genre prediction, and digit prediction).

(b) *How many instances are there in total (of each type, if appropriate)?*
We describe each dataset's statistics in detail in Appendix C.2. We chose datasets to span small, medium, and large sizes. The smallest dataset contains $1,460$ instances (and training a model takes roughly a few minutes on a single GPU) while the largest one contains $306,245$ instances (and training a model takes roughly 2 weeks on a single GPU). This enables accessibility for researchers with limited computational resources, while also allowing for large-scale studies of multimodal datasets and models.

(c) *Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).*
Each of the datasets is collected in different ways that we detail in Appendix C.2. To summarize, each dataset consists of samples from a larger set since it is impossible to include all videos/stock data/medical data/robotics data in the world. Each dataset is collected with the aim to be representative of the entire population.

(d) *What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.*
We describe in detail the raw data and processed features for each dataset in Appendix C.2. To summarize, MULTIBENCH contains both raw modality data as well as processed data with predefined feature extractors following current work.

(e) *Is there a label or target associated with each instance? If so, please provide a description.*
We describe in detail the labels for each dataset in Appendix C.2. To summarize, MULTIBENCH contains 6 research areas with a total of 15 prediction tasks spanning affect recognition, robot manipulation, stock prediction, design interface, action recognition, movie genre prediction, and digit prediction.

(f) *Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.*
No, all datasets are provided in full. For robustness tests, we do inject noise and imperfections into each dataset to simulate the performance of machine learning models on real-world imperfections (see Appendix D.3 for details).

(g) *Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.*

We describe in detail the relationships between modalities for each dataset in Appendix C.2.

(h) *Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.*
Yes, MULTIBENCH provides a data loading pipeline that directly loads train, validation, and test splits according to current work. We provide these details for each dataset in Appendix C.2.

(i) *Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.*
We do not know of any errors in each of the datasets included in MULTIBENCH. However, we will always be on the lookout for potential issues and update them via https://cmu-multicomp-lab.github.io/multibench/ and https://github.com/pliang279/MultiBench.

(j) *Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.*
Most of the datasets in MULTIBENCH have been collected, stored, processed, and are self-contained. There are some datasets that depend on external resources which we explain below:

   i. MIMIC: We depend on the original dataset to be hosted on https://mimic.physionet.org/gettingstarted/access/. Unfortunately, since we are not allowed to redistribute the raw data and users need to complete training to access the raw data, we are unable to provide a self-contained version of the MIMIC dataset. We are currently planning to add several new multimodal datasets in the healthcare domain that can be self-contained after appropriate de-identification.
   ii. Finance: Yahoo Finance prohibits the redistribution of their data. We depend on the original data to be hosted on Yahoo Finance and provide automated downloading and preprocessing scripts for the datasets based on `pandas-datareader`, which has original code at https://github.com/pydata/pandas-datareader/blob/master/pandas_datareader/yahoo/components.py

(k) *Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.*
From the authors of MIMIC [78]: "The project was approved by the Institutional Review Boards of Beth Israel Deaconess Medical Center (Boston, MA) and the Massachusetts Institute of Technology (Cambridge, MA). Requirement for individual patient consent was waived because the project did not impact clinical care and all protected health information was de-identified."
To the best of our knowledge, all other datasets do not contain confidential data and are publicly available for research purposes.

(l) *Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.*
We reviewed the datasets and found no offensive content. While there are clearly expressions of highly negative sentiment or strong displays of anger and disgust in the affective computing videos, there are no offensive words used or personal attacks recorded in the video. All videos are related to movie or product reviews, TED talks, and TV shows.

(m) *Does the dataset relate to people? If not, you may skip the remaining questions in this section.*
Yes, the healthcare, affective computing, and Kinetics (multimedia) datasets relate to people. The other datasets in MULTIBENCH do not.

(n) *Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.*
The following datasets relate to people:

i. Affective computing: These datasets do not identify any subpopulations in their modeling decisions. However, the raw data comes in the form of videos publicly available and free to download from YouTube. Sub-population and demographic information can be inferred from these raw videos.
ii. MIMIC: According to the authors [78]: "The median age of adult patients is 65.8 years and 55.9% patients are male."
iii. Kinetics: This dataset does not identify any subpopulations. However, the raw data comes in the form of videos publicly available and free to download from YouTube. Sub-population and demographic information can be inferred from these raw videos.

(o) *Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.*
The following datasets relate to people:

i. Affective computing: One can see the person in the raw video, but the dataset contains no personal information. We do not explicitly use information regarding gender, ethnicity, identity, or video identifier in online sources. All pre-extracted features are non easily invertible and only rely on general visual or audio features such as the presence of a smile or magnitude of voice [181, 183].
ii. MIMIC: The MIMIC dataset has been rigorously de-identified in accordance with Health Insurance Portability and Accountability Act (HIPAA) such that all possible personal information has been removed from the dataset. Removed personal information includes patient name, telephone number, address, and dates. Dates of birth for patients aged over 89 were shifted to obscure their true age. Please refer to Appendix C.2.2 for de-identification details. Again, we emphasize that any multimodal models trained to perform prediction should only be used for scientific study and should not in any way be used for real-world prediction.
iii. Kinetics: One can see the person in the raw video, but the dataset does not contain direct personal information. We do not explicitly use information regarding gender, ethnicity, identity, or video identifier in online sources.

(p) *Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.*
MULTIBENCH contains datasets with financial and healthcare data. However, all these datasets are publicly available for research purposes. Healthcare data (MIMIC) has been rigorously de-identified in accordance with the Health Insurance Portability and Accountability Act (HIPAA) such that all possible personal information (patient name, telephone number, address, and dates, date of birth) has been removed from the dataset. Please refer to Appendix C.2.2 for de-identification details.

(q) *Any other comments?*
No.

3. **Collection Process**

(a) *How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.*
We include the collection process for each dataset in Appendix C.2.

(b) *What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?*
We include these details in Appendix C.2.

(c) *If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?*
We include sampling methods for each dataset in Appendix C.2.

(d) *Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?*
We include annotation details for each dataset in Appendix C.2.

(e) *Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.*
We include timeframes for each dataset in Appendix C.2.

(f) *Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.*
From the authors of MIMIC [78]: "The project was approved by the Institutional Review Boards of Beth Israel Deaconess Medical Center (Boston, MA) and the Massachusetts Institute of Technology (Cambridge, MA). Requirement for individual patient consent was waived because the project did not impact clinical care and all protected health information was de-identified."

(g) *Does the dataset relate to people? If not, you may skip the remainder of the questions in this section.*
Yes, the healthcare, affective computing, and Kinetics (multimedia) datasets relate to people. The other datasets in MULTIBENCH do not.

(h) *Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?*
Affective computing and Kinetics datasets are collected from YouTube videos that follow the creative commons license and follow fair use guidelines of YouTube. According to the authors for the MIMIC dataset [78]: "Data was downloaded from several sources, including archives from critical care information systems, hospital electronic health record databases, and Social Security Administration Death Master File."

(i) *Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how the notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.*
Affective computing and Kinetics datasets are collected from YouTube videos that follow the creative commons license and follow fair use guidelines of YouTube. This is the standard way for content creators to grant someone else permission to use and redistribute their work. According to the authors for the MIMIC dataset [78]: "The project was approved by the Institutional Review Boards of Beth Israel Deaconess Medical Center (Boston, MA) and the Massachusetts Institute of Technology (Cambridge, MA). Requirement for individual patient consent was waived because the project did not impact clinical care and all protected health information was de-identified."

(j) *Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.*
Affective computing and Kinetics datasets are collected from YouTube videos that follow the creative commons license and follow fair use guidelines of YouTube which allows content creators to grant someone else permission to use and redistribute their work. According to the authors for the MIMIC dataset [78]: "Requirement for individual patient consent was waived because the project did not impact clinical care and all protected health information was de-identified."

(k) *If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).*
N/A.

(l) *Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description*

of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.
N/A.

(m) *Any other comments?*
N/A.

4. **Preprocessing/cleaning/labeling**

(a) *Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.*
Yes, we followed the convention in prior research for any preprocessing done to the datasets. We explain these steps in Appendix C.2.

(b) *Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.*
Yes, we include the raw data in MULTIBENCH in addition to the preprocessed features. The raw data (usually in the form of raw text, videos, audio, time series etc) are useful for users to perform their own feature extraction and also for robustness tests on raw data itself (e.g., imperfections in the raw text through spelling errors and missing words). There are certain cases where we are not allowed to distribute the raw data: for MIMIC where users must undergo training to download the raw data, and for finance datasets where Yahoo Finance is publicly available but does not allow redistribution of raw data. For both of these datasets, we provide automated download and preprocessing scripts once the raw data is downloaded through the correct procedure by each user (see details in Appendix C.2).

(c) *Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.*
Yes, we provided all links and references to preprocessing steps in Appendix C.2.

(d) *Any other comments?*
No.

5. **Uses**

(a) *Has the dataset been used for any tasks already? If so, please provide a description.*
Yes, MULTIBENCH contains several datasets that have been used in the multimodal ML community. We provide links to the original repositories of each dataset and their original citations in Appendix C.2.

(b) *Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.*
We provide links to the original repositories of each dataset and their original citations in Appendix C.2. We also include references to general multimodal methods implemented in MULTIZOO in Appendix E. Many of these methods have been tested by their original authors on a small subset of datasets in MULTIBENCH. In addition to these references, the leading authors maintain a reading list on topics in multimodal ML at [98] which contains links to papers, datasets, code, academic courses, conferences, and workshops relevant to the multimodal ML community.

(c) *What (other) tasks could the dataset be used for?*
In addition to building multimodal models for the prediction tasks, datasets in MULTIBENCH can also be used for:

i. Unsupervised learning across multimodal data/unsupervised pre-training of multimodal models.
ii. Interpreting relationships between modalities.
iii. Designing models for robustness to noisy and missing modalities.
iv. Investigating alignment between modalities.
v. Other multimodal tasks including but not limited to: co-learning, translation, retrieval, and grounding [10].

(d) *Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other*

undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?
We are careful to outline all possible risks associated with each dataset in Appendix C.2 and also in our broader impact statement (Appendix A). We acknowledge that there could be risks regarding the privacy and security of data, as well as the real-world deployment of these methods whenever human-centric data is involved (e.g., in healthcare, affective computing, and multimedia). We discussed data demographics in the previous section and it should be taken into consideration when making claims regarding the generalization of models to new users. We also emphasize that these multimodal datasets and methods should only be used for research purposes and not for actual real-world deployment until research can sufficiently verify their safety. Finally, we are carefully working with domain experts towards better understanding biases in these multimodal datasets and models as well as their real-world safety.

(e) *Are there tasks for which the dataset should not be used? If so, please provide a description.*
Yes, we emphasize that all multimodal models trained to perform prediction on these datasets should not in any way be used to harm individuals and should only be used as a scientific study. They should not be deemed safe for real-world deployment. In particular, the models used to make predictions of affective states, human actions, health indicators, and financial indicators are particularly sensitive and should not be used to inform any real-world decisions. All results must only be used as a scientific study of machine learning methods. See more details in Appendix A.

(f) *Any other comments?*
No.

6. **Distribution**

(a) *Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.*
Yes, the benchmark will be distributed to the public research community for theoreticians and practitioners to experiment on multimodal data.

(b) *How will the dataset be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?*
We plan to distribute MULTIBENCH via our public GitHub: https://github.com/pliang279/MultiBench. We also include a landing website page on https://cmu-multicomp-lab.github.io/multibench/ that includes an introduction to the benchmark, links to the relevant papers on multimodal datasets and algorithms, and a public leaderboard to keep track of current progress on these multimodal tasks.

(c) *When will the dataset be distributed?*
The dataset is currently available for use.

(d) *Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.*
We release the benchmark and code under an MIT license: see https://github.com/pliang279/MultiBench/blob/main/LICENSE, which allows for sharing and distribution of the code for research purposes. Each of the datasets in MULTIBENCH has their own licenses which we detail in Appendix C.2.

(e) *Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.*
Yes, MULTIBENCH brings together a collection of several existing datasets in the multimodal research that were built by their individual authors who have original licenses for these datasets. We only included the datasets with licenses that allow for redistribution (MIT or Creative Commons license) and are freely downloadable for research purposes. We detailed all dataset licenses in Appendix C.2.

(f) *Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.*
We are not aware of any such restrictions.

(g) *Any other comments?*
No.

7. **Maintenance**

(a) *Who is supporting/hosting/maintaining the dataset?*
The dataset is supported and hosted by the team of authors at CMU. The team will also lead the maintenance and expansion of MULTIBENCH. The team will also work with the other collaborators on the paper who are domain experts in each research area MULTIBENCH covers, such as robotics, HCI, healthcare, and finance.

(b) *How can the owner/curator/manager of the dataset be contacted (e.g., email address)?*
We provide all contact addresses at https://cmu-multicomp-lab.github.io/multibench/.

(c) *Is there an erratum? If so, please provide a link or other access point.*
All erratum and updates to the dataset will be tracked via GitHub commit histories at https://github.com/pliang279/MultiBench. We will also provide updates via our landing page https://cmu-multicomp-lab.github.io/multibench/.

(d) *Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?*
Yes, we plan for long-term maintenance and expansion of the dataset. All erratum and updates to the dataset will be tracked via GitHub commit histories at https://github.com/pliang279/MultiBench. We will also provide updates via our landing page https://cmu-multicomp-lab.github.io/multibench/. Please refer to Appendix C.5 for details.

(e) *If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.*
The individuals in question were not notified about the data collection. For YouTube videos, they are released under a creative commons license which is the standard way for content creators to grant someone else permission to use and redistribute their work. According to the authors for the MIMIC dataset [78]: "The project was approved by the Institutional Review Boards of Beth Israel Deaconess Medical Center (Boston, MA) and the Massachusetts Institute of Technology (Cambridge, MA). Requirement for individual patient consent was waived because the project did not impact clinical care and all protected health information was de-identified."

(f) *Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.*
Yes, we will maintain a GitHub history for all updates and older versions of datasets and code in MULTIBENCH.

(g) *If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.*
Yes, we will create a system where users can create pull requests on GitHub to include their datasets and models. The authors will verify that the additions are in the scope of multimodal learning and do not break the current experimental code. We will work with these authors to ensure that their data and algorithms can be included in MULTIBENCH.

(h) *Any other comments?*
No.

C.4 Benchmark Distribution

We plan to distribute the MULTIBENCH benchmark via our public GitHub: https://github.com/pliang279/MultiBench. We also include a landing website page on https://cmu-multicomp-lab.github.io/multibench/ that includes an introduction to the benchmark, links to the relevant papers on multimodal datasets and algorithms, and a public leaderboard to keep track of current progress on these multimodal tasks.

The GitHub and webpage will also allow feedback from the research community in suggesting and adding new datasets and algorithms. Finally, we plan to include a list of planned future updates to MULTIBENCH on the webpage along with their target release dates.

C.5 Hosting and Maintenance

We have a long-term plan to continue the expansion and maintenance of MULTIBENCH. Here we summarize the main directions we plan to expand towards and leave details and other areas of future work to Appendix I.

- Maintenance: MULTIBENCH will be continuously hosted via GitHub which provides stable access to code and a landing page website. We guarantee that MULTIBENCH will be available for a long time through our distribution channels. The authors themselves are also actively working on multimodal learning in affective computing, robotics, healthcare, human-computer interaction, and multimedia. The authors are also involved in efforts in applying multimodal machine learning to finance. As a result of these long-term collaborative research efforts, the authors will continue to maintain and expand on the datasets and code provided in MULTIBENCH.
- Expansion of datasets: We plan to include more datasets for multimodal fusion as well as more research areas in multimodal learning such as retrieval, question answering, grounding, and reinforcement learning. While these research areas are very different, we hope that insights in multimodal representations can be shared across them.
- Expansion of evaluation: To enable holistic evaluation, we plan to build on top of our metrics by adding robustness to distribution shift, uncertainty measures, tests for fairness and social biases, as well as labels/metrics for interpretable multimodal learning.
- Expansion of datasets: We plan to encourage students taking the multimodal machine learning course at CMU (https://cmu-multicomp-lab.github.io/mmml-course/fall2020/) to use the benchmark and add their proposed datasets and models to it.
- Expansion of methods: The authors currently collect a very up-to-date reading list of core multimodal papers https://github.com/pliang279/awesome-multimodal-ml and plan to continuously update MULTIZOO with new multimodal methods proposed by the community.

C.6 Author Statement

The authors carefully reviewed the information present in this document. To the best of our knowledge, the datasets in MULTIBENCH can be used for research purposes, following the methodology and licenses described in the dataset section (Appendix C.2).

C.7 License

Each of the datasets included in MULTIBENCH includes their own licenses which we detail in Appendix C.2.. We release all preprocessing code across all datasets using the MIT license. All other codes for multimodal algorithms in MULTIZOO as well as evaluation scripts, are also released via an MIT license: see https://github.com/pliang279/MultiBench/blob/main/LICENSE, which allows for sharing and distribution of the code for research purposes.

C.8 Metadata

We have included structured metadata for MULTIBENCH on our landing page: https://cmu-multicomp-lab.github.io/multibench/.

C.9 Persistence of MULTIBENCH

MULTIBENCH is publicly hosted on https://github.com/pliang279/MultiBench. For larger datasets that cannot be uploaded to GitHub, we plan to upload the processed dataset to CMU Box. We are still exploring the best options for sharing large datasets. Users need to download these processed datasets, place them into a correct folder, and run the MULTIBENCH data loader and machine learning pipeline.

D MULTIBENCH Evaluation Protocol

To enable holistic evaluation, MULTIBENCH offers a comprehensive evaluation methodology to assess (1) generalization across domains and modalities, (2) complexity during training and inference, and (3) robustness to noisy and missing modalities: We describe the evaluation protocol for each desiderata in detail in each of the following subsections:

D.1 Performance

MULTIBENCH provides standardized evaluation using metrics designed for each dataset, ranging from MSE and MAE for regression to accuracy, micro & macro F1-score, and AUPRC for classification on each dataset. To assess for generalization, we compute the variance of a particular model's performance across all datasets in MULTIBENCH on which it is tested. We split these results on multiple datasets into *in-domain* datasets and *out-domain* datasets. *In-domain* datasets refer to model performance on datasets that it was initially proposed and tested on, while *out-domain* datasets refer to model performance on the remaining datasets. Comparing out-domain vs in-domain performance, as well as variance in performance across datasets as a whole, allow us to summarize the generalization statistics of each multimodal model.

D.2 Complexity

Modern ML research, unfortunately, causes significant impacts to energy consumption [142], a phenomenon often exacerbated in processing high-dimensional multimodal data. As a step towards quantifying energy complexity and recommending lightweight multimodal models, MULTIBENCH records the amount of information taken in bits (i.e., data size), number of model parameters, and time and memory resources required during the entire training process. To enforce consistency, the training time measured for all models on each dataset is run on the same CPUs and GPUs. We report training memory by measuring peak memory usage of the python process during the entire training process using python `memory_profiler` toolkit (https://pypi.org/project/memory-profiler/). When counting the number of parameters when training a model, we only count the parameters in persistent modules during training and does not count the ephemeral networks or modules created in the middle of the training process (such as the networks trained for determining weights in GRADBLEND or the fusion architectures created as part of the architecture search process in MFAS).

In addition to training time and resources, real-world models may need to be small and compact to run on mobile devices [131]. To account for this, MULTIBENCH also records *inference time and parameters*. We report inference time by measuring the time it takes for the trained model to complete inference on the entire test set of the dataset. In some cases, only parts of the parameters used in training are counted towards the inference parameters (for example, the parameters in decoders of MVAE and MFM are part of training parameters but not part of inference parameters).

D.3 Robustness to Imperfect Data

Real-world multimodal data is often imperfect as a result of missing entries, noise corruption, or missing modalities entirely. For example, multimodal dialogue systems trained on acted TV shows are susceptible to poor performance when deployed in the real world where users might be less expressive in using facial gestures. This calls for robust models that can still make accurate predictions despite only having access to a (possibly noisy [101]) subset of signals [123]. To standardize efforts in evaluating the robustness of multimodal models, MULTIBENCH includes the following robustness tests as part of the evaluation:

D.3.1 Modality-specific Imperfections

Modality-specific imperfections are independently applied to each modality taking into account the unique noise topologies in that source of data (i.e., flips and crops of images, natural misspellings in text, abbreviations in spoken audio). We describe all the modality-specific imperfections we implement in MULTIBENCH in the following:

Language: Imperfections in the language modality can occur at various granularities spanning the character, word, phrase, and sentence levels. With reference to [15], many of these imperfections occur at the raw text data level and are usually results of spelling errors on a QWERTY keyboard

as well as abbreviations in written, typed, and spoken text. Given a word w of length n and a fixed probability $p \in (0, 1)$, we implement the following language-specific imperfections:

1. Spelling errors: note that spelling mistakes are different from intentionally changed word forms (e.g. abbreviation used in instant messaging service) since they are unintentional [144]. We simulate typos by replacing each letter with a letter having an adjacent position on a QWERTY keyboard with probability p.
2. Short message noise: Short Message Service (SMS) data usually include intentional corruptions of words and phrases like abbreviations, phonetic substitutions, omission of characters and words, and dialectal and informal usages [144]. We implement the following:
   (a) Simulate sticky keys: given a number m, choosing m letters of a word randomly to repeat with probability p.
   (b) Simulate quick typing: given a number m, choosing m letters of a word randomly to omit with probability p.
3. Random permutation of letters: swapping adjacent two letters is a common natural noise when typing quickly [15]. Random permutation of the entire word or the majority of letters is a form of synthetic noise. We implement the following:
   (a) Swap two random adjacent letters (except for the first and the last letter) with probability p.
   (b) Permute the middle chunk of a word: denote the middle chunk (all letters except the first and the last letter) as $w[1 : n]$, with probability p, produce a permutation f with the first and last letter fixed, i.e. $f(0) = 0, f(n) = n$. The shuffled word is w' with $w'[f(i)] = w[i]$ for all $i \in [n]$.

Image: Given a RGB image $\mathbf{X} \in \mathbb{Z}^{W \times H \times 3}$ where W and H are the height and width of the image, let R, G, B be the $W \times H$ matrices of three color channels. We implement the following robustness tests in the image modality:

1. Noises in digital images: various noises are naturally prevalent in digital images during image acquisition, coding, transmission, and processing steps [19]. We implement the following:
   (a) Gaussian/electronic noise that normalizes histogram with respect to the gray values. We add Gaussian noise as a $W \times H$ matrix with each entry following Gaussian distribution $\mathcal{N}(0, p)$.
   (b) Impulse valued/salt-and-pepper noise that has dark pixels in bright regions and bright pixels in dark regions. To add salt-and-pepper noise, for each pixel $x \in \mathbf{X}$, we convert $x = 0$ (white) or $x = 255$ (black) into a dead pixel with uniform distribution with probability p.
   (c) Periodic noise such that it looks like some repeating patterns are exposed on top of the affected image. We add periodic noise by exposing the original image to periodic patterns with probability p.
2. Color errors:
   (a) Convert the image to grayscale: $0.3R + 0.59G + 0.11B$ with probability p.
   (b) Decrease the contrast with probability p.
   (c) Negate the color: let **X'** be the inverted image then $\forall i \in [W], j \in [H], k \in [3], \mathbf{X}'(i, j, k) = 255 - \mathbf{X}[i, j, k]$ with probability p.
   (d) Change the white-balance by increasing/decreasing the temperature with probability p.
   (e) Colorize the image with probability p.
3. Flips, crops, and rotations:
   (a) Horizontal flipping with probability p.
   (b) Color space transformation - isolating a single color channel and changing brightness etc with probability p.
   (c) Random cropping changes with probability p.
   (d) Rotate the image by random angle $\in [20, 40]$ with probability p.
   (e) Translation of images to the left, right, up, or down with probability p.

Most of these transformations are achieved with the Python Imaging Library (PIL).

Video: We treat video data as a time series of images. For each image in the video, we apply the image-specific robustness tests as described above. In addition, we also apply the following tests to simulate imperfections in time-series data:

1. Random drop: dropping the datapoint at random time step with probability p.
2. Structured drop: given a time step t, m consecutive time steps with at least one nonzero signal are dropped with probability p.

Audio: Audio is typically represented as a time-series signal. Noises are primarily caused by imperfections in the recording device, which can cause static Gaussian noise to be added to the recorded temporal waveform at random time steps, background noise to be picked up at higher magnitudes, and certain time steps (or consecutive time steps) to be dropped from the recording. We implement the following unimodal noises in the audio modality:

1. Additive white Gaussian noise: given an array of length N of a sampled audio segment, we add white gaussian noise, which is an array of N with each entry following a normal distribution with mean 0 and standard deviation p.

In addition to these imperfections applied at a single time step, we also apply the following across the entire time-series signal:

1. Random drop: dropping the datapoint at random time step with probability p.
2. Structured drop: given a time step t, m consecutive time steps with at least one nonzero signal are dropped with probability p.

Time-series data consists of a sequence with a time-dimension (a sequence of data points indexed by time). Following Liang et al., [101], we implement the following types of noise and missing values in time-series data:

1. White noise added independently at every time step (noise sampled from zero-mean Gaussian with standard deviation p).
2. Random drop: dropping the datapoint at random time step with probability p.
3. Structured drop: given a time step t, m consecutive time steps across modalities are dropped with probability p.

Optical flow: We treat optical flow in a similar manner as time-series data and implement the same robustness tests.

Force and proprioception sensors: We also treat these sensors in robotics as time-series data with a key difference - we add noise/drop time steps at a higher frequency since force and proprioception sensors often record data at a higher frequency.

Tabular data takes the form of rows, each of which contains information about some feature (e.g., age,). We define the following robustness tests on tabular data:

1. Random drops of elements from the table with probability p.
2. Random swaps elements in the table with probability p.

Sets are data instances where the collection of input elements satisfy permutation invariance, which is in contrast to fixed dimensional vectors that are commonplace in machine learning on images, text, and audio. The key difference between sets and tabular data is that each element in the set is often assumed to be from the same distribution (e.g., a point cloud is a set of 3D coordinates). We define the following types of noise on an input set modality:

1. Random dropping of elements from the set with probability p.
2. Adding noise to elements of the set with noise sampled from zero-mean Gaussian with standard deviation p.

D.3.2 Multimodal Imperfections

Multimodal imperfections capture correlations in imperfections across modalities (e.g., missing modalities [123], or a chunk of time missing in multimodal time-series data [101]). These represent settings where data collection across modalities is correlated rather than independent.

1. Correlated noise: adding noise to all modalities with probability p, where noise is defined according to the aforementioned modality-specific noises.
2. Correlated drop: dropping all modalities with probability p, where dropping patterns are defined according to the aforementioned modality-specific drops.

3. Temporal drop: in the case of temporal modalities recorded in parallel (e.g., video, audio, and text recorded across time; financial time-series data recorded across days), we perform correlated drops across all modalities at random time steps with probability p.
4. Structured temporal drop: we extend temporal drop such that given a time step t, we perform temporal drop on m consecutive time steps with probability p.
5. Missing modalities: dropping an entire modality with probability p.

D.3.3 Robustness Measure

We train the model on clean training data and evaluate it under increasing levels of noise added only to test data. To simulate realistic noise and imperfections in test data, we follow the modality-specific and multimodal imperfections as described above. Given a multimodal dataset with M modalities, this allows us to create $M + 1$ partitions of imperfect test datasets: one partition of increasing noise levels for modality-specific imperfections within each modality (which gives a total of M partitions) and one partition of multimodal imperfections across all modalities. For datasets where it is not possible to create multimodal imperfections due to the lack of a shared dimension (e.g., image and text datasets typically do not share any correlated dimension, but multimodal time-series datasets share an underlying time dimension), we implement the first M modality-specific imperfections which results in M imperfect data partitions.

A qualitative visualization: Given each test partition, we take a unimodal or multimodal model trained on clean data and plot model performance on the y-axis as increasing levels of noise is added to the test data, on a range of 0 (no noise) to 1 (complete noise) along the x-axis. This allows us to visually inspect the robustness of each model as increasing imperfections are added to the test data. Visually, a robust model should maintain high accuracy (or low MSE) as much as possible despite increasing levels of noise.

A quantitative metric: While the visualization technique above allows one to compare the robustness of several multimodal models across the same dataset, it does not allow us to aggregate robustness performance across the broad range of datasets and tasks in MULTIBENCH. To design such a metric, we extend the quantitative robustness measures proposed in Taori et al., [149] to deal with multimodal imperfections across a range of imperfection levels $\sigma \in [0.0, 1.0]$.

We begin by reviewing the example proposed in Taori et al., [149]: suppose we are given two models f_1 and f_2, where accuracy $\text{acc}_{\text{clean}}(f_1) = 0.8$, $\text{acc}_{\text{noisy}}(f_2) = 0.75$ (i.e., a 5% drop in accuracy from the imperfections), and $\text{acc}_{\text{clean}}(f_2) = 0.9$, $\text{acc}_{\text{noisy}}(f_2) = 0.76$ (a 14% drop). Model f_2 has higher accuracy on the noisy test set, but overall sees a drop of 14% from the clean to the noisy test set. In contrast, f_1 starts off with a lower accuracy but sees only a 5% drop. To capture both these desiderata (i.e., having higher accuracy at *all* levels and *lower drops* in accuracy), Taori et al., [149] introduce two notions of robustness: relative and effective robustness.

Relative robustness directly measures accuracy under imperfection. A model with higher relative robustness would display higher accuracy at *all* levels of imperfection compared to a baseline model. We measure the relative robustness of all multimodal models as compared to a baseline LF (simple late fusion with concatenation) method since that is the most basic method tested on all datasets. We compute relative robustness of a model f using the formula

$$\tau(f) = \int_{\sigma} \text{acc}_{\sigma}(f) - \text{acc}_{\sigma}(\text{LF})\, \mathrm{d}\sigma, \tag{1}$$

which essentially measures the area between two performance-imperfection curves as imperfection levels σ increase from 0.0 to 1.0 (we compute a discrete approximation to the integral).

Effective robustness measures the rate of accuracy drops as imperfection levels increase. However, to reliably measure the rate of accuracy drops, one must remove the confounding variable brought by differences in initial accuracies on clean test data. Taori et al., [149] therefore propose to measure whether a model can offer higher accuracy on the noisy test set *beyond what is expected from having higher accuracy on the original test set*. Taori et al., use a log-linear fit on the set of (accuracy on noisy test data, accuracy on clean test data) points across a range of models trained on ImageNet to measure the expected accuracy on noisy test data given a new model's performance on clean test data. Graphically, effective robustness then corresponds to a model's performance on noisy test data lying above the linear trendline. Similar to relative robustness, we measure the effective robustness of multimodal models relative to the accuracy trend of the LF baseline, which we denote as β_{LF}. We

compute effective robustness of a model f using the formula

$$\rho(f) = \int_{\sigma} \mathrm{acc}_{\sigma}(f) - \beta_{\mathrm{LF}}\left(\mathrm{acc}_{0.0}(f)\right)\ \mathrm{d}\sigma, \tag{2}$$

which essentially measures the area between the performance-imperfection curve of model f and a *shifted* performance-imperfection curve of the LF baseline (shifted to match the initial accuracy of model f at imperfection level 0.0). A model with higher effective robustness should lie above this shifted accuracy curve at all imperfection levels σ. Again, we compute a discrete approximation to the integral.

Overall, a robust multimodal model should obtain both high relative and effective robustness.

D.4 Aggregating Measures Across Datasets and Tasks

MULTIBENCH benefits from benchmarking multimodal models across a diverse set of datasets, modalities, and tasks. While it is useful to analyze methods on a single dataset in isolation, it is also useful to assess the generalization and failure modes of methods across multiple datasets. Therefore, we need a way to reliably summarize the above metrics (performance, complexity, and robustness) across datasets despite their being on vastly different scales (e.g., accuracy for different numbers of categories) and orders (e.g., accuracy vs RMSE). We find that min-max normalization of results per dataset into a $0-1$ scale (where min and max are appropriately reversed for RMSE/MSE metrics) before averaging across datasets gives a reliable indicator of overall performance across multiple datasets.

Table 3: MULTIZOO provides a standardized implementation of the following multimodal methods to enable accessibility for new researchers and reproducibility of results. These approaches span advances in data processing, fusion paradigms, optimization objectives, and training procedures. We choose these approaches since they offer complementary perspectives towards tacking the fundamental challenges in multimodal fusion: (1) aligning signals across modalities at the right granularity, (2) learning complementary information across aligned signals, and (3) maintaining robustness in the presence of noisy and missing modalities.

Category	Method	Alignment	Complementarity	Robustness
Data	WORDALIGN [26]	✓	✗	✗
Model	EF, LF [10]	✗	✓	✗
	TF [179], LRTF [106]	✗	✓	✗
	MI-MATRIX, MI-VECTOR, MI-SCALAR [77]	✗	✓	✗
	NL GATE [167]	✗	✓	✗
	MULT [154]	✓	✓	✗
	MFAS [122]	✗	✓	✗
Objective	CCA [7]	✓	✗	✗
	REFNET [135]	✓	✗	✗
	MFM [155]	✗	✓	✗
	MVAE [168]	✗	✓	✗
	MCTN [123]	✗	✗	✓
Training	GRADBLEND [167]	✗	✓	✓
	RMFE [53]	✗	✓	✓

E MULTIZOO: A Zoo of Multimodal Algorithms

In this section, we provide more details into our choice of standardizing multimodal representation learning as well as the implementation of our standardized library. In each category, we carefully describe the algorithm, motivate its effect in tackling one of the core challenges in section B.2, and provide references to the original code that we adapted to include in MULTIZOO.

E.1 Selection of Algorithms in MULTIZOO

We begin by discussing our choices of algorithms in MULTIZOO. We consulted with domain experts in each of the application areas to select methods that satisfy the following properties:

1. *Diversity in areas:* We chose algorithms that present novel perspectives across a suite of machine learning research domains spanning data preprocessing, fusion paradigms, optimization objectives, and training procedures.
2. *Coverage of technical challenges:* Each of the algorithms selected in MULTIZOO are chosen because they provide unique perspectives to the technical challenges in multimodal learning as elucidated in Appendix B.2. In Table 3, we provided a coarse attempt in categorizing each of the technical challenges in multimodal learning. As a result, we did not include too many methods in any category (e.g., multiple methods that are based on model architectures that tackle similar challenges of learning complementary information). Even within the same category and within those tackling the same technical challenge, we attempted to select ones that were fundamentally different (e.g., architectures based on domain knowledge, general-purpose Transformers, and architecture search).
3. *SOTA on a particular dataset:* For each dataset chosen in MULTIBENCH, we aim to include the model that currently achieves state-of-the-art performance on that dataset. This allows us to assess the best performing model *within* the same domain of the dataset, as well as the best performing model *outside* the domain of the dataset.
4. *Community expansion:* Any set of initial methods that we will choose will represent only a small sample of the powerful multimodal methods out there. We will encourage community participation in expanding the methods in MULTIZOO and encourage researchers to implement new methods using a similar modular structure to reduce confounding factors, enable standardized sharing of code, and ensure reproducibility in results.

E.2 Data Preprocessing

Temporal alignment: As a preprocessing step, performing temporal alignment [26] has been shown to help tackle the multimodal alignment problem in the case of time-series data. This approach makes an implicit assumption on the temporal granularity of the modalities (e.g., at the level of words for text) and aligns information from the remaining modalities to the same

temporal granularity. We call this approach WORDALIGN [26] and apply it to temporal data with text being one of the modalities. We use the temporal alignment provided in https://github.com/A2Zadeh/CMU-MultimodalSDK. Specifically, it performs alignment at the granularity of words. Given a sentence with words $w_1, ..., w_T$ each annotated with their start and end times $(s_1, e_1), (s_2, e_2), ..., (s_T, e_T)$, word-level alignment takes the non-text modality features (which are typically extracted at a higher frequency) and averages them during the intervals $e_1 - s_1, e_2 - s_2, ..., e_T - s_T$. This results in a text sequence of T words alongside aligned non-text modality sequences of T time-steps as well.

E.3 Fusion Paradigms

Early and late fusion have been the de-facto first-approach when tackling new multimodal problems. Early fusion performs concatenation at the input data level before using a suitable prediction model (i.e., $\mathbf{z}_{\text{mm}} = [\mathbf{x}_1, \mathbf{x}_2]$) and late fusion applies suitable unimodal models to each modality to obtain their feature representations, concatenates these features, and defines a classifier to the label (i.e., $\mathbf{z}_{\text{mm}} = [\mathbf{z}_1, \mathbf{z}_2]$) [10]. MULTIZOO includes their implementations denoted as EF and LF respectively. Since these are basic building blocks in the multimodal learning field, we implement them ourselves.

Tensors are specifically designed to tackle the multimodal complementarity challenge by explicitly capturing higher-order interactions across modalities [179]. Given unimodal representations $\mathbf{z}_1, \mathbf{z}_2$, a multimodal tensor representation is defined as $\mathbf{z}_{\text{mm}} = \begin{bmatrix}\mathbf{z}_1 \\ 1\end{bmatrix} \otimes \begin{bmatrix}\mathbf{z}_2 \\ 1\end{bmatrix}$ where $\otimes$ denotes an outer product. However, computing tensor products is expensive since their dimension scales exponentially with the number of modalities. Several efficient variants have been proposed to approximate expensive full tensor products with cheaper variants while maintaining performance [71, 101, 106]. MULTIZOO includes Tensor Fusion (TF) [179] as well as approximate Low-rank Tensor Fusion (LRTF) [106].

We use the Tensor Fusion implementation in https://github.com/Justin1904/TensorFusionNetworks and the Low-rank Tensor Fusion implementation in https://github.com/Justin1904/Low-rank-Multimodal-Fusion. As future work, we also plan to include more expressive higher-order tensor fusion methods [71].

Multiplicative Interactions (MI) further generalize tensor products to include learnable parameters that capture the interactions between streams of information [77]. In its most general form, MI defines a bilinear product $\mathbf{z}_{\text{mm}} = \mathbf{z}_1 \mathbb{W} \mathbf{z}_2 + \mathbf{z}_1^\top \mathbf{U} + \mathbf{V}\mathbf{z}_2 + \mathbf{b}$ where $\mathbb{W}, \mathbf{U}, \mathbf{Z}$, and $\mathbf{b}$ are trainable parameters. By appropriately constraining the rank and structure of these parameters, MI recovers HyperNetworks [61] (unconstrained parameters resulting in a matrix output), Feature-wise linear modulation (FiLM) [120, 188] (diagonal parameters resulting in vector output), and Sigmoid units [37] (scalar parameters resulting in scalar output). MULTIZOO includes all 3 as MI-MATRIX, MI-VECTOR, and MI-SCALAR respectively.

Since code was not released for the Multiplicative Interactions paper [77], we implemented the MI layer ourselves. We also referred to the implementation of Feature-wise linear modulation (FiLM) [120] from https://github.com/ethanjperez/film and added it as a module in MULTIBENCH, which we call FILM. While MI-VECTOR (i.e., diagonal parameters in a MI layer which results in a vector output) corresponds to the most basic implementation of FILM, the original FILM layer uses multiple non-linear layers instead of a single linear transformation in MI-VECTOR which has been shown to improve performance [120].

Gated attention models are prevalent in learning combinations of two representations that dynamically change for every input [25, 167, 171]. Its general form can be written as $\mathbf{z}_{\text{mm}} = \mathbf{z}_1 \odot h(\mathbf{z}_2)$, where h represents a function with sigmoid activation and $\odot$ denotes the element-wise product. The output $h(\mathbf{z}_2)$ is commonly referred to as "attention weights" learned from $\mathbf{z}_2$ used to attend on $\mathbf{z}_1$.

We implement the Query-Key-Value mechanism as NL GATE as proposed in [167] by referring to the implementation of in https://github.com/facebookresearch/VMZ. This attention mechanism is conceptually similar to the MI-VECTOR case above but recent work has explored more expressive forms of h such as using a Query-Key-Value mechanism [167] or several fully-connected layers [25] rather than a linear transformation in MI-VECTOR.

Temporal attention models are useful in tackling the challenge of multimodal alignment and complementarity. Transformer models [158] have been shown to be useful for temporal multimodal data by automatically aligning and capturing complementary features at different time-steps [154,

174]. We include the Multimodal Transformer (MULT) [154] which uses a Crossmodal Transformer block that uses $\mathbf{z}_1$ to attend to $\mathbf{z}_2$ (and vice-versa), before concatenating both representations to obtain $\mathbf{z}_{\text{mm}} = [\mathbf{z}_{1\to2}, \mathbf{z}_{2\to1}] = [\text{CM}(\mathbf{z}_1, \mathbf{z}_2), \text{CM}(\mathbf{z}_2, \mathbf{z}_1)]$.

We use the public implementation available at https://github.com/yaohungt/Multimodal-Transformer which includes a basic crossmodal transformer block designed for 2 modalities. To extend this to 3 modalities, the crossmodal transformer block is repeated across all 3 sets of modality pairs (i.e., $\mathbf{z}_{\text{mm}} = [\mathbf{z}_{1\to2}, \mathbf{z}_{2\to1}, \mathbf{z}_{1\to3}, \mathbf{z}_{3\to1}, \mathbf{z}_{2\to3}, \mathbf{z}_{3\to2}]$). While this is still computationally feasible for 3 modalities such as the language, video, and audio datasets that MULT was originally designed for, this quickly becomes intractable for problems involving more than 3 modalities. To adapt MULT for the financial prediction task involving more than 10 modalities, we cluster all modalities into 3 groups based on similarities in their data and perform early fusion on the data within each cluster before applying MULT only on the 3 clusters of modalities. While MULT is a strong model based on performance, it poses scalability issues that should be the subject of future work (i.e., since the number of cross-modal attention blocks grows quadratically with the number of modalities).

Architecture search: Finally, instead of hand-designing multimodal architectures, several approaches define a set of atomic neural operations (e.g., linear transformation, activation, attention, etc.) and use architecture search to automatically learn the best order of these operations for a given multimodal task [122, 173]. We focus on the more general approach, MFAS [122], designed for language and vision datasets.

We adapt the implementation from https://github.com/juanmanpr/mfas. While this approach is categorized under innovations in model architecture (since it primarily targets better architectures for multimodal fusion), its code in the MULTIZOO toolkit is implemented under training structures, since architecture search requires an outer loop to learn model architectures over multiple inner supervised learning loops that train an individual model architecture. Therefore, we are unable to integrate MFAS directly with the basic supervised learning training loops like we do for the other fusion paradigms described above.

E.4 Optimization Objectives

In addition to the standard supervised losses (e.g., cross-entropy for classification, MSE/MAE for regression), several proposed methods have proposed new optimization objectives based on:

Prediction-level alignment: There has been extensive research in defining objective functions to tackle the challenge of multimodal alignment: capturing a representation space where semantically similar concepts from different modalities are close together. While primarily useful for cross-modal retrieval [104, 187], recent work has also shown its utility in learning representations for prediction [9, 33, 91, 151]. These alignment objectives have been applied at both prediction and feature levels. In the former, we implement Canonical Correlation Analysis (CCA) [7, 166], which computes $\mathcal{L}_{\text{CCA}} = \text{corr}\left(g_1(\mathbf{z}_1), g_2(\mathbf{z}_2)\right)$ where g_1, g_2 are auxiliary classifiers mapping each unimodal representation to the label. This method corresponds to prediction-level alignment since they aim to learn representations of each modality that agree on the label, as measured by the correlation of label predictions made by each modality across a batch of samples.

We refer to the paper that most closely implements CCA-based alignment for multimodal data (specifically directly testing on the CMU-MOSI dataset) [145]. Since the authors did not release their code, we implemented it from scratch with reference to CCA implementations from https://github.com/Michaelvll/DeepCCA and https://github.com/VahidooX/DeepCCA.

Feature-level alignment: In the latter, contrastive learning has emerged as a popular approach that brings similar concepts close in feature space and different concepts far away [33, 91, 151]. MULTIZOO includes REFNET [135] which includes a self-supervised contrastive loss between unimodal representations $\mathbf{z}_1, \mathbf{z}_2$ and the multimodal representation $\mathbf{z}_{\text{mm}}$, i.e., $\mathcal{L}_{\text{contrast}} = 1 - \cos(\mathbf{z}_{\text{mm}}, g_1(\mathbf{z}_1)) + 1 - \cos(\mathbf{z}_{\text{mm}}, g_2(\mathbf{z}_2))$ where g_1, g_2 is an auxiliary layer mapping each modality's representation into the joint multimodal space. The intuition here is that the unimodal representations $\mathbf{z}_1, \mathbf{z}_2$ and the multimodal representation $\mathbf{z}_{\text{mm}}$ should be aligned in the multimodal feature space as measured by cosine similarity. While the original REFNET method does not use negative samples, closely related work in multi-view contrastive learning has extended this idea to use negative samples which is more closely in line with recent work in contrastive learning [151].

Since they did not release code, we implement REFNET ourselves on top of current supervised learning modules in MULTIZOO.

Reconstruction objectives: Methods based on generative-discriminative models (e.g., VAEs) include an objective to reconstruct the input (or some part of the input) [91, 155]. These have been shown to better preserve task-relevant information learned in the representation, especially in settings with sparse supervised signals such as robotics [91] and long videos [155]. We include the Multimodal Factorized Model (MFM) [155] which is a general approach that learns a representation $\mathbf{z}_{\text{mm}}$ that can reconstruct input data $\mathbf{x}_1, \mathbf{x}_2$ while also predicting the label. The multimodal representation is a concatenation of factorized representations $\mathbf{z}_1$, $\mathbf{z}_2$, ..., $\mathbf{z}_M$, and $\mathbf{z}_y$.

Since MFM optimizes a variational lower-bound to the log likelihood, the overall objective consists of 3 terms - generative, discriminative, and prior regularization:

$$\min_{f_i, f_{\text{mm}}, g_i, g_y} \mathbf{E}_{P_{\mathbf{x}_{1:M}, \mathbf{y}}} \mathbf{E}_{f_1(\mathbf{z}_1|\mathbf{x}_1)} \cdots \mathbf{E}_{f_M(\mathbf{z}_M|\mathbf{x}_M)} \mathbf{E}_{f_{\text{mm}}(\mathbf{z}_y|\mathbf{x}_{1:M})} \left[\sum_{i=1}^{M} \|\mathbf{x}_i, g_i(\mathbf{z}_i, \mathbf{z}_{\mathbf{y}})\|_2 + \ell\left(\mathbf{y}, g_y(\mathbf{z}_{\mathbf{y}})\right) \right] + \lambda \text{MMD}(Q_{\mathbf{z}}, P_{\mathbf{z}}), \tag{3}$$

where f_i's are encoders from each modality to representations, f_{mm} is a multimodal encoder to the joint representation $\mathbf{z}_y$, g_i's are decoders from latent representations back into input data, and g_y is a classification head to the label. The final MMD term is a regularizer to bring the representations close to a unit Gaussian prior. The multimodal encoder f_{mm} in MFM can be instantiated with any multimodal model from section 3.2 (e.g., learning $\mathbf{z}_y$ via tensors and adding a term to reconstruct input data). We use the public implementation in https://github.com/pliang279/factorized, which uses a temporal attention model as f_{mm} for multimodal time-series data. For the remaining experiments we replace f_{mm} with a simple late fusion but also run some experiments with multimodal methods that are state-of-the-art in each domain.

Improving robustness: These approaches modify the objective function to account for robustness to noisy [101] or missing [89, 111, 123] modalities. MULTIZOO includes MCTN [123] which uses cycle-consistent translation to predict the noisy/missing modality from present ones. The key insight is that a joint representation between modalities $\mathbf{x}_1$ and $\mathbf{x}_2$ can be learned by using $\mathbf{x}_1$ to predict $\mathbf{x}_2$, in a vein similar to machine translation or image/text style transfer. MCTN defines a cyclic translation path $\mathbf{x}_1 \rightarrow \mathbf{z}_{\text{mm}} \rightarrow \hat{\mathbf{x}}_2 \rightarrow \mathbf{z}_{\text{mm}} \rightarrow \hat{\mathbf{x}}_1$ and adds additional reconstruction losses $\mathcal{L}_{\text{rec}} = \|\mathbf{x}_1 - \hat{\mathbf{x}}_1\|_2 + \|\mathbf{x}_2 - \hat{\mathbf{x}}_2\|_2$ on top of the supervised learning loss. The representations $\mathbf{z}_{\text{mm}}$ learned via translation are then used to predict the label. Surprisingly, the model needs to take in only $\mathbf{x}_1$ at test time and is therefore robust to all levels of noise or missingness in $\mathbf{x}_2$.

E.5 Training Procedures

Improving generalization: Recent work has found that directly training a multimodal model with all modalities using supervised learning is sub-optimal since different modalities overfit and generalize at different rates. MULTIZOO includes an approach to solve this, called Gradient Blending (GRADBLEND), that computes generalization statistics for each modality to determine their weights during multimodal fusion [167]. We use the implementation in https://github.com/facebookresearch/VMZ and modify it to be part of the MULTIZOO training structures.

We also include a similar work, Regularization by Maximizing Functional Entropies (RMFE), which uses functional entropy to balance the contribution of each modality to the classification result [53]. We use the public implementation from https://github.com/itaigat/removing-bias-in-multi-modal-classifiers.

E.6 Domain-specific Methods

Finally, we also implemented several domain-specific methods that had been applied to each domain. These include sensor fusion [91] and Kalman filtering [90] for robotics, and the multimodal Refiner network [135] for multimedia experiments. We refer the reader to the respective papers for algorithmic details.

F Integrating MULTIBENCH and MULTIZOO: A Brief Tutorial

MULTIBENCH is available via our public GitHub: https://github.com/pliang279/MultiBench. We also include a landing website page on https://cmu-multicomp-lab.github.io/multibench/ that includes an introduction to the benchmark, links to the relevant papers on multimodal datasets and algorithms, and a public leaderboard to keep track of current progress on these multimodal tasks. In this section, we provide more details for the loading of datasets ML pipeline provided by MULTIBENCH. We also describe the modular implementation of multimodal models in MULTIZOO and provide several code examples to illustrate its usage.

F.1 Reading the Dataset

We provide scripts for reading each dataset supported by MULTIZOO at `dataset/[dataset_name]/get_data.py` in the repository. For each dataset, the user will need to first follow downloading and preprocessing instructions documented in Section C.2 or in the comments of the `get_data.py`. The python script contains a function (usually called `get_dataloader`) that takes in required arguments (such as the location of the preprocessed dataset or compressed data, etc) and it will output a tuple of three PyTorch Dataloader objects for train, valid, and test split of the dataset respectively. You can feed these dataloaders directly into training structures in MULTIZOO.

F.2 Unimodal Models

In addition to the multimodal models described in Appendix E that are the main subject of study in this area, each dataset and modality typically also requires an initial processing stage either through feature extraction (see Appendix C.2 for initial feature extraction done on each dataset) and/or unimodal models on raw data/extracted features.

To standardize the implementation of unimodal models, MULTIZOO includes an implementation of several standard unimodal models that we encountered when running experiments on the diverse range of datasets and modalities in MULTIBENCH. Each unimodal model is implemented as a function class that takes in either raw data or extracted features from a modality and returns a unimodal representation tensor after applying the function. MULTIZOO includes the following unimodal methods:

1. MULTI-LAYER PERCEPTRONS form the building blocks of many deep learning methods and are generally suitable for any modality that has undergone feature extraction into a vector that does not require any more processing with inductive biases. Their general structure means that they can be flexibly adapted for the tabular, set, and image, and text (e.g., see Deep Averaging Network [76]) modalities. They have also been used as a starting point for force and proprioception sensors in robotics if data does not come in the form of time-series [91].
2. CONVOLUTIONAL NETWORKS [87] are typically used over the image modality. They are also used on the audio modality if an initial preprocessing step of converting raw audio to spectrograms is used.
3. RESNETS [66] are an improvement over ConvNets to enable training of deeper models and have been used extensively for images and audio spectrograms.
4. RECURRENT NETWORKS [134], GRUS [29], and LSTMS [69] are suitable for temporal data in the form of text, video, audio, and time-series modalities.
5. TRANSFORMERS [158] have recently emerged as a strong alternative to recurrent models by using self-attention rather than an accumulative memory. They are also suitable for text, video, audio, and time-series modalities. We also implemented recently proposed VISION TRANSFORMERS [44] that adapt Transformer models for image classification as well.
6. DEEP SETS [184] was proposed as a permutation-invariant method for machine learning on sets, and was shown to outperform prior methods such as MLPs that are sensitive to the permutation of elements.
7. Finally, we also included several domain-specific methods that we encountered as we were accumulating the datasets in MULTIBENCH. Some of these methods include MAXOUT networks [58] used for MM-IMDB [8] and CAUSAL CONVOLUTION [157] for the high-frequency force sensors used in robotics datasets [91, 90].

F.3 Multimodal Models

MULTIZOO includes an implementation of all multimodal methods described in Appendix E. Each multimodal method (i.e., fusion paradigm) is implemented as a Pytorch Module class taking in unimodal tensors and returning final multimodal representation vectors. We implemented several common fusion modules, such as Concatenation, Early-Concatenation (i.e., concatenate in input space), Stack, FilM, Multiplicative-Interactions (MI), Tensor Fusion, LRTF, NL-gate, and more described in Appendix E. When the training algorithm requires non-standard multimodal representations (e.g., more than one vector output from fusion module) or the unimodal encoders produce non-standard unimodal representations (i.e., not a single vector representation), special fusion modules will be needed in these situations. For example, we wrote a `roboticsConcat` module that performs concatenation for the VISION&TOUCH dataset due to its non-standard unimodal encoder output. We also have special fusion modules for optimization objectives or training structures such as MVAE, MFAS, and GRADBLEND. The design of modular fusion modules gives flexibility in model design, as users can reuse a previous fusion module directly in most cases but can also write their own special fusion modules easily.

F.4 Classification Head

Finally, MULTIZOO includes flexible implementations of classification heads that take in the multimodal representation and return a label either directly (perhaps with some activation) for regression or a softmax over classes for classification.

F.5 Optimization Objectives

The optimization objectives are modules that take in the classification or regression result produced by the model and the ground-truth (as well as other necessary inputs if applicable) and return a loss that can be used to optimize the model based on the desired objective. In most methods we simply use `torch.nn.CrossEntropyLoss` as the objective for classification tasks and `torch.nn.MSELoss` as the objective for regression tasks. However, in certain training structures, special objectives are required. For example, MULTIZOO includes implementations of objective functions such as weighted reconstruction loss and ELBO loss used in reconstruction-based methods MFM and MVAE, and there are also implementations of alignment-based objectives such as CCA and contrastive learning. The final optimization objective returns a weighted sum of these prediction objectives and auxiliary objectives, where the user is free to specify these weights as hyperparameters.

F.6 Training Structures

Training Structures are the main body of MULTIZOO programs. All other modules (unimodal models, fusion paradigms, optimization objectives, classification heads, etc) can be seen as exchangeable plugins to these training structures. The training structure determines the main training algorithm, with the most common one being `supervised_learning` (training unimodal, multimodal, and classification parameters directly for a task-specific supervised learning objective).

More advanced methods may change this training structure either through additional optimization objectives (MVAE [168], MFM [155]) or via extensions of supervised learning through dynamic weighting of modalities (GRADBLEND [167]) or an outer architecture search training loop (MFAS [122]). Each of these methods, therefore, have their own training structure module.

These interchangeable plugin modules give a lot of flexibility in adapting each training structure to new tasks. For example, for the experiments described in Section G, the methods that are primarily based on different fusion paradigms (i.e., EF, LF, TF, LRTF, MI, NL-GATE, MULT etc all use the same training structure (`supervised_learning`) with different plugin fusion modules (and different unimodal encoders and heads based on datasets and tasks). Similarly, while most of these more advanced training structures were originally paired with a simple LF model in their original papers, our modular implementation makes it possible to combine advances in fusion paradigms with training structures in future work.

F.7 Performance Evaluation

We standardize evaluation using metrics designed for each dataset, ranging from MSE and MAE for regression to accuracy, micro macro F1-score, and AUPRC for classification. We use the standard PyTorch and scikit-learn implementations of these performance metrics.

Algorithm 2 PyTorch code integrating MULTIBENCH datasets and MULTIZOO models.

```
from datasets.get_data import get_dataloader
from unimodals.common_models import ResNet, Transformer
from fusions.common_fusions import MultInteractions
from training_structures.gradient_blend import train, test

# loading Multimodal IMDB dataset
traindata, validdata, testdata = get_dataloader('multimodal_imdb')
out_channels = 3
# defining ResNet and Transformer unimodal encoders
encoders = [ResNet(in_channels=1, out_channels, layers=5),
            Transformer(in_channels=1, out_channels, layers=3)]
# defining a Multiplicative Interactions fusion layer
fusion = MultInteractions([out_channels*8, out_channels*32], out_channels*32, 'matrix')
classifier = MLP(out_channels*32, 100, labels=23)
# training using Gradient Blend algorithm
model = train(encoders, fusion, classifier, traindata, validdata,
        epochs=100, optimtype=torch.optim.SGD, lr=0.01, weight_decay=0.0001)
# testing
performance, complexity, robustness = test(model, testdata)
```

F.8 Complexity Evaluation

We report training memory by measuring peak memory usage of the python process during the entire training process using python `memory_profiler` toolkit (https://pypi.org/project/memory-profiler/). When counting the number of parameters when training a model, we only count the parameters in persistent modules during training and does not count the ephemeral networks or modules created in the middle of the training process (such as the networks trained for determining weights in GRADBLEND or the fusion architectures created as part of the architecture search process in MFAS).

F.9 Robustness Evaluation

For robustness experiments, modality-specific and multimodal imperfections are implemented as modules. A separate version of data loader is created for each dataset to test robustness, which adds custom unimodal or multimodal imperfections of increasing noise levels $\sigma \in [0, 1]$ to the original clean test set. A testing module is also provided specifically for robustness experiments, which evaluates the model on increasing levels of noisy test datasets and prints out the metrics for visualization. In this way, MULTIZOO allows highly modular data loading and robustness evaluation that requires minimal modification to the regular training and testing workflow.

MULTIZOO includes evaluation protocols summarizing these robustness results. It includes visualization functions of the performance-imperfection curves across datasets in MULTIBENCH. We also implemented relative and effective robustness as two quantitative metrics for robustness evaluation. For relative robustness, we approximate the area under the performance-imperfection curves for each model across MULTIBENCH datasets. For effective robustness, we take the performance-imperfection curve of LF evaluated on the same dataset equalized for initial accuracy on clean test data. For both metrics, we normalized performance across all models evaluated on the same dataset.

F.10 Code Snippets

In Algorithm 2, we show a sample code snippet in Python that loads a dataset from MULTIBENCH (Appendix C.2), defines the unimodal and multimodal architectures, optimization objectives, and training procedures (Appendix E), before running the evaluation protocol (Appendix D). Our MULTIZOO toolkit is easy to use and trains entire multimodal models in less than 10 lines of code. By standardizing the implementation of each module and disentangling the individual effects of models, optimizations, and training, MULTIZOO ensures accessibility and reproducibility of its multimodal algorithms.

Table 4: Table of hyperparameters for prediction on affective computing dataset.

Component	Model	Parameter	Value
GRU Encoder	GRU	Input sizes	$[5, 20, 35, 74, 300, 704]$
		Hidden sizes	$[32, 32, 64, 128, 512, 1024]$
		Num of layers	1 or 2
		Dropout	0.0 or 0.1
Transformer Encoder [158]	Transformer [158]	Input sizes	$[5, 20, 35, 74, 300, 704]$
		Hidden sizes	$[5, 10, 20, 40, 40, 50]$
		Num heads	2 or 3
		Dropout	0.2
Head	MLP	Input sizes	$[5, 20, 32, 64, 128, 256]$
		Hidden sizes	$[5, 20, 32, 64, 128, 256]$
		Num layers	2
		Dropout	0.2
MCTN [123] Encoder	GRU	Input sizes	300
		Hidden sizes	$[32, 64]$
		Num of layers	1 or 2
		Dropout	0.0 or 0.1
MCTN [123] Decoder	GRU	Input sizes	$[32, 64]$
		Hidden sizes	300
		Num of layers	1 or 2
		Dropout	0.0 or 0.1
MCTN [123] Seq2Seq	GRU+GRU	teaching ratio	0.5
		Embed sizes	32
		$\mu_{t_1}, \mu_c, \mu_{t_2}$	0.01
Fusion	LRTF [106]	Num ranks	64
		Output sizes	128
	MI-MATRIX [77]	Hidden size	128
	MULT [156]	Hidden size	40
		Num heads	8 or 10
Training		Loss	MAE or Cross Entropy
		Batch size	32
		Seq Length	50 or 20
		Num epochs	100 or 300
		Early stop	True
		Patience	$[8, 20]$
		Activation	ReLU
		Optimizer	AdamW
		Weight Decay	1×10^{-4}
		Learning rate	1×10^{-4}

G Experimental Setup

In this section, we provide additional details of the experimental setup. All experiments were conducted on a server with 4× Nvidia GTX 980 Ti GPUs, 5× Nvidia Tesla P40 GPUs, 2× Nvidia Tesla K40c GPUs, 4× Nvidia TITAN X GPUs, 1× Tesla T4 GPU, and 1× Tesla V100 GPU. The server also contained 32× Intel(R) Xeon(R) CPU (E5 – 2670, 2.60GHz).

G.1 Affective Computing

Hyperparameters: We show the hyperparameters used for models on datasets in the Affective Computing domain in Table 4. For each dataset we tune the following hyperparameters selected from the following ranges: the learning rate is selected between 0.00001 to 0.001 and set to be 0.0001 in the beginning; Early stopping is applied with patience 8 to 20 before overfitting happens; The input sizes and hidden sizes vary according to the different modalities and datasets. The μ_{t_0}, μ_c, and μ_{t_1} hyperparameters in MCTN [123] is tuned between 0.005 to 0.1. The sequence length varies from 20 to 50. Only punchline sentences (target sentences) are used in UR-FUNNY [64] and MUSTARD [24] following the original papers.

Hyperparameters were selected based on performance on the validation set. For models that had been previously proposed and tested on these datasets, we use the same hyperparameters as those reported in their paper or public code.

All experiments were repeated 10 times and a mean and standard deviation was computed.

G.2 Healthcare

We show the hyperparameters used for models on datasets in the Healthcare domain in Table 5. The unimodal architectures follow the original paper that created this partition of MIMIC [129], then we tune the following hyperparameters selected from the following ranges: Learning rate is tuned between 0.1 and 0.0001; the number of epochs is selected based on when overfitting happens; for hyperparameters specific to architectures or training structures (such as GRADBLEND, MFAS), we followed the same configuration as the original papers where these methods are proposed.

All experiments were repeated 10 times and a mean and standard deviation was computed.

G.3 Robotics

We show the hyperparameters used for MUJOCO PUSH in Table 6 and VISION&TOUCH in Table 7.

For MUJOCO PUSH, we follow hyperparameters and preprocessing in the original paper [90]. Unimodal modules follow the original hyperparameters assigned to the input modality.

For VISION&TOUCH, we follow hyperparameters in the original paper [91] for all unimodal modules as well as Sensor Fusion (which is the method proposed in [91]).

All other hyperparameters were selected based on performance on the validation set. For models that had been previously proposed and tested on these datasets, we use the same hyperparameters as those reported in their paper or public code. The original VISION&TOUCH dataset did not have a unique test dataset, so we report their best performance on the validation set instead.

All experiments were repeated 10 times and a mean and standard deviation was computed.

G.4 Finance

We show the hyperparameters used for models on datasets in the Finance domain in Table 8. For each dataset, we tune the following hyperparameters selected from the following ranges: Hidden/embed dim (4 – 512), Transformer/MULT layers (1 – 4), Transformer/MULT heads (1 – 4), epochs (1 – 32), and batch size (4 – 128). Hyperparameters were selected based on performance on the validation set. Note that this dataset overfits quickly when model complexity is increased; several hyperparameters are kept small for this reason.

All experiments were repeated 10 times and a mean and standard deviation was computed.

G.5 HCI

We show the hyperparameters used for models on the ENRICO dataset in the HCI domain in Table 9. We tune the learning rate by starting from 10^{-4}, the value reported in the original paper [93]. We searched in a range between 10^{-2} and 10^{-6} and found that 10^{-5} led to the best performance. We tested hidden dimension sizes from 8 to 128 and found that a size of 16 was sufficient for the unimodal encoders. Note that this dataset is small and overfits quickly when model complexity is increased. We minimized the risk of overfitting by keeping several hyperparameters (*e.g.,* hidden dim) small. For more information, refer to the dataset preprocessing section for ENRICO.

All experiments were repeated 10 times and a mean and standard deviation was computed.

G.6 Multimedia

We show the hyperparameters used for models on datasets in the Multimedia domain in Tables 10, 11, 12.

For AV-MNIST, used the same LeNet unimodal encoders following current work [161]. We tuned learning rates between 0.1 and 0.001. The default batch size is 40, although it can be changed in some methods (such as CCA) to make sure the methods work as intended; the number of epochs is selected based on when overfitting happens; for hyperparameters specific to architectures or training

structures (such as GRADBLEND, MFAS), we followed the same configuration as the original papers where these methods are proposed.

For MM-IMDB, used the same MaxoutLinear unimodal encoders following current work [8]. Learning rates were tuned between 0.1 and 0.001 except for unimodal training. The default batch size is 128 while that for CCA is 800 to make sure the methods work as intended. The number of epochs was selected based on early stopping with patience equal to 7, which means if the macro F1 on the validation set did not improve for 7 epochs, training was stopped early.

For KINETICS, we use a ResNet-LSTM for the visual modality encoder and the architectures described by Wang et al [167] for the rest of the models. We use a learning rate of 0.0001, batch size of 16, and 15 epochs for the small dataset experiments. For the large dataset experiments, we used the setup described by Wang et al [167].

Hyperparameters were selected based on performance on the validation set. For models that had been previously proposed and tested on these datasets, we use the same hyperparameters as those reported in their paper or public code.

All experiments were repeated 10 times and a mean and standard deviation was computed.

Table 5: Table of hyperparameters for prediction on MIMIC dataset in the healthcare domain.

Component	Model	Parameter	Value
Static Encoder	2-layer MLP	Hidden sizes	$[10, 10]$
		Activation	LeakyReLU(0.2)
Static Decoder	2-layer MLP	Layer sizes	$[200, 40, 5]$
		Activation	LeakyReLU(0.2)
Time Series Encoder	GRU	Hidden dim	30
Time Series Decoder	GRU	Hidden dim	30
Classification Head	2-Layer MLP	Hidden size	40
		Activation	LeakyReLU(0.2)
Fusion	LRTF [106]	Output dim	100
		Ranks	40
	NL-Gate [167]	thw-dim/c-dim/tf-dim	$24/30/10$
		key linear	$[10, 300]$
		value linear	$[10, 300]$
	MI-Matrix [77]	output dim	100
Training	Unimodal, LF, LRTF, MI-Matrix, NL-gate	Loss	Cross Entropy
		Batch size	40
		Num epochs	20
		Optimizer	RMSprop
		Learning rate	0.001
	GRADBLEND [167]	Loss	Cross Entropy
		Batch size	40
		Num epochs	300
		Optimizer	SGD
		Learning Rate	0.005
		GB-epoch	20
		v-rate	0.8
		finetune epoch	25
	MVAE [168]	Loss	Cross Entropy + ELBO
		Batch size	40
		Num epochs	30
		Optimizer	Adam
		Learning Rate	0.001
		Cross Entropy Weight	2.0
		Latent Representation Fusion	ProductOfExpert
	MFM [155]	Loss	Cross Entropy + Reconstruction(MSE)
		Batch size	40
		Num epochs	30
		Optimizer	Adam
		Learning Rate	0.001
		Recon Loss Modality Weights	$[1, 1]$
		Cross Entropy Weight	2.0
		Intermediate Modules	MLPs $[200, 100, 100]$, $[200, 100, 100]$, $[400, 100, 100]$
	MFAS [122]	Batch size	32
		Epochs/search iters	$3/3/6$
		Num samples/surrogates per epoch	$15/50$
		η max/min/Ti/Tm	$10^{-3}/10^{-6}/1/2$
		Temperature init/final/decay	$10.0/0.2/4.0$
		Max progression level	4
		Surrogate learning rate	0.001
		Surrogate hidden size	100
		Surrogate embedding size	100
		Search space	$(3, 3, 2)$
		Optimizer	Adam
		Representation Size	16

Table 6: Table of hyperparameters for prediction on MuJoCo Push dataset in the robotics domain.

Component	Model	Parameter	Value
Pos Encoder	Linear	Hidden sizes	$[64, 64, 64$ (residual)$]$
Sensors Encoder	Linear	Hidden sizes	$[64, 64, 64$ (residual)$]$
Image Encoder	CNN	Filter sizes	$[5, 3, 3, 3, 3]$
		Num filters	$[32, 32, 32, 16, 8]$
		Filter strides	1
		Filter padding	$[2, 1, 1, 1, 1]$
Control Encoder	Linear	Hidden sizes	$[64, 64, 64$ (residual)$]$
Fusion	Early Fusion & Unimodal LSTM	Hidden size	512
		Num layers	2
	Late Fusion LSTM	Hidden size	256
		Num layers	1
	MulT [156]	Embed size	64
		Num heads	4
Classification Head	Linear	Hidden size	64
Training		Loss	Mean Squared Error
		Batch size	32
		Num epochs	20
		Activation	ReLU
		Optimizer	Adam
		Learning rate	10^{-5}

Table 7: Table of hyperparameters for prediction on Vision&Touch dataset in the robotics domain.

Component	Model	Parameter	Value
Image Encoder	CNN	Filter sizes	$[7, 5, 5, 3, 3, 3]$
		Num filters	$[16, 32, 64, 64, 128, 128]$
		Filter strides	$[2, 2, 2, 2, 2, 2]$
		Filter padding	Same
Force Encoder	Causal Convolution [157]	Filter sizes	$[2, 2, 2, 2, 2]$
		Num filters	$[16, 32, 64, 128, 256]$
		Filter strides	$[2, 2, 2, 2, 2]$
		Filter padding	1
Proprio Encoder	Linear	Hidden sizes	$[32, 64, 128, 256]$
Depth Encoder	CNN	Filter sizes	$[3, 3, 4, 3, 3, 3]$
		Num filters	$[32, 64, 64, 64, 128, 128]$
		Filter strides	$[2, 2, 2, 2, 2, 2]$
		Filter padding	Same
Action Encoder	Linear	Hidden sizes	$[32, 32]$
Classification Head	2-Layer MLP	Hidden size	128
		Activation	LeakyReLU(0.2)
Fusion	LRTF [106]	Output dim	200
		Ranks	40
	Sensor Fusion [91]	z-dim	128
Training		Loss	Contact: Cross Entropy End-Effector: MSE
		Batch size	64
		Num epochs	Sensor Fusion: 50 LRTF: 35; Others: 15
		Optimizer	Adam
		Learning rate	Contact: 10^{-4} End-Effector: 5×10^{-4}
	RefNet [135]	Loss	Cross Entropy + Contrast
		Batch size	40
		Optimizer/Learning Rate	Adam / 0.0005
		Refiner	MLP$(1056, 2000, 65760)$
		Self Loss Weight	0.0001

Table 8: Table of hyperparameters for stock prediction on finance datasets (we use the same hyperparameters on all 3 datasets: STOCKS-F&B, STOCKS-HEALTH, and STOCKS-TECH).

Model	**Parameter**	**Value**	
Unimodal & Early Fusion LSTM	Hidden dim	128	
Late Fusion LSTM	Hidden dim	16	
TRANSFORMER [?]	Embed dim	9	
	Num heads	3	
	Layers	3	
MULT [154]	Embed dim	9	
	Num heads	3	
	Layers	3	
GRADBLEND [167] LSTM	Hidden dim	128	
Training	Loss	Mean Squared Error	
	Batch size	16	
	Max seq length	500	
	Activation	ReLU	
	Optimizer	Adam	
	Learning rate	10^{-3}	
	Num epochs	Unimodal, EF	2
		LF, Transformer, MULT, GRADBLEND	4

Table 9: Table of hyperparameters for prediction on ENRICO dataset in the HCI domain.

Model	**Parameter**	**Value**
Unimodal	Hidden dim	16
Late Fusion	Hidden dim	32
GRADBLEND [167]	Hidden dim	32
REFNET [135]	Hidden dim	32
MI-Matrix [77]	Hidden dim	32
	Input dims	16, 16
Tensor Matrix	Hidden dim	32
	Input dims	16, 16
LRTF [106]	Hidden dim	32
	Input dims	16, 16
	Rank	20
CCA [145]	Hidden dim	32
Training	Loss	Class-weighted Cross Entropy
	Batch size	32
	Activation	ReLU
	Dropout	0.2
	Optimizer	Adam
	Learning rate	10^{-5}
	Num epochs	50

Table 10: Table of hyperparameters for prediction on AV-MNIST dataset in the multimedia domain.

Component	Model	Parameter	Value
Image Encoder	LeNet-3	Filter Sizes	$[5, 3, 3, 3]$
		Num Filters	$[6, 12, 24, 48]$
		Filter Strides / Filter Paddings	$[1, 1, 1, 1]$ /$[2, 1, 1, 1]$
		Max Pooling	$[2, 2, 2, 2]$
Image Decoder	DeLeNet-3	Filter Sizes	$[4, 4, 4, 8]$
		Num Filters	$[24, 12, 6, 3]$
		Filter Strides / Filter Paddings	$[2, 2, 2, 4]$/$[1, 1, 1, 1]$
Audio Encoder	LeNet-5	Filter Sizes	$[5, 3, 3, 3, 3, 3]$
		Num Filters	$[6, 12, 24, 48, 96, 192]$
		Filter Strides / Filter Paddings	$[1, 1, 1, 1, 1, 1]$/$[2, 1, 1, 1, 1, 1]$
		Max Pooling	$[2, 2, 2, 2, 2, 2]$
Audio Decoder	DeLeNet-5	Filter Sizes	$[4, 4, 4, 4, 4, 8]$
		Num Filters	$[96, 48, 24, 12, 6, 3]$
		Filter Strides / Filter Paddings	$[2, 2, 2, 2, 2, 4]$/$[1, 1, 1, 1, 1, 1]$
Classification Head	2-Layer MLP	Hidden size	100
		Activation	LeakyReLU(0.2)
Fusion	LRTF [106]	Output dim	120
		Ranks	40
	MI-Matrix [77]	output dim	240
Training	Unimodal, LF, LRTF, MI-Matrix	Loss	Cross Entropy
		Batch size	40
		Num epochs	LRTF: 30, Others: 25
		Optimizer/Learning rate/weight decay	SGD/0.05/0.0001
	GRADBLEND [167]	Loss	Cross Entropy
		Batch size	40
		Num epochs	300
		Optimizer/Learning rate	SGD/0.05
		GB-epoch/finetune-epoch	10/25
		v-rate	0.8
	MVAE [168]	Loss	Cross Entropy + ELBO
		Batch size	40
		Num epochs	20
		Optimizer/Learning rate	Adam/0.001
		Cross Entropy Weight	2.0
		Latent Representation Fusion	ProductOfExpert
	MFM [155]	Loss	Cross Entropy + Reconstruction(MSE)
		Batch size	40
		Num epochs	25
		Optimizer/Learning rate	Adam/0.001
		Recon Loss Modality Weights	$[1, 1]$
		Cross Entropy Weight	2.0
		Intermediate Modules	MLPs $[200, 100, 100]$, $[200, 100, 100]$, $[400, 100, 100]$
	MFAS [122]	Batch size	32
		Main epochs/search iters/epochs per model	$3/3/6$
		Num samples/surrogates per epoch	$15/50$
		η max/min/Ti/Tm	$10^{-3}/10^{-6}/\ 1/2$
		Temperature init/final/decay	$10.0/0.2/4.0$
		Max progression level	4
		Surrogate learning rate	0.001
		Surrogate hidden/embedding size	$100/100$
		Search space	$(3, 5, 2)$
		Optimizer	Adam
		Representation Size	16
	CCA [145]	Batch size	800
		Loss	CCALoss
		Optimizer/Learning Rate/Weight Decay	AdamW/ $0.01/0.01$
	REFNET [135]	Loss	Cross Entropy + Contrast
		Batch size	40
		Optimizer/Learning Rate	SGD / 0.05
		Refiner	MLP$(384, 1000, 13328)$
		Self Loss Weight	0.1

Table 11: Table of hyperparameters for prediction on MM-IMDB dataset in the multimedia domain.

Component	Model	Parameter	Value
Text Encoder	2-Layer MaxoutMLP	Hidden size	512
		Output dim	128/256/512
		MLP num	2
Image Encoder	2-Layer MaxoutMLP	Hidden size	1024
		Output dim	128/256/512
		MLP num	2
Classification Head	Linear		
	2-Layer MLP	Hidden size	512
		Activation	ReLU
	2-Layer Maxout_Linear	Hidden size	512
		MLP num	2
Fusion	Concatenate		
	LRTF [106]	Output dim	512
		Ranks	128
	MI-Matrix [77]	output dim	1024
Training	Unimodal, EF, LF, LRTF, MI-Matrix	Loss	Binary Cross Entropy
		Batch size	128
		Num epochs	Text: 125, Image: 25, LF:5, EF/LRTF:15, MI-Matrix:20
		Optimizer	AdamW
		Learning rate	Unimodal: 0.0001, EF: 0.04, LF/LRTF/MI-Matrix: 0.008
		Weight decay	0.01
	CCA [145]	Loss	Binary Cross Entropy + CCA
		CCA weight	0.001
		Batch size	800
		Num epochs	20
		Optimizer	AdamW
		Learning rate	0.01
		Weight decay	0.01
	RMFE [53]	Loss	Binary Cross Entropy + Regularization
		Regularization weight	$1e-10$
		Batch size	128
		Num epochs	10
		Optimizer	AdamW
		Learning rate	0.01
		Weight decay	0.01
	REFNET [135]	Loss	Binary Cross Entropy + Contrast + Self-supervised
		Contrast weight	0.0001
		Self-supervised weight	0.1
		Batch size	128
		Num epochs	10
		Optimizer	AdamW
		Learning rate	0.01
		Weight decay	0.01
	MFM [155]	Loss	Binary Cross Entropy + Reconstruction(MSE)
		Batch size	128
		Num epochs	10
		Optimizer	Adam
		Learning rate	0.005
		Recon Loss Modality Weight	$[1, 1]$
		Cross Entropy Weight	2.0
		Intermediate Modules	MLP $[512, 256, 256]$ MLP $[512, 256, 256]$ MLP $[1024, 512, 256]$

Table 12: Table of hyperparameters for prediction on KINETICS dataset in the multimedia domain.

Component	Model	Parameter	Value
Video Encoder	ResNet [66] + LSTM	ResNet Version	18-layer
		LSTM Hidden size	64
Audio Encoder	ResNet [66] + 2-Layer MLP	ResNet Version	50-layer
		MLP hidden size	200
		MLP output size	64
		MLP activation	ReLU
Classification Head	Linear		
	2-Layer MLP	Hidden size	200
		Activation	ReLU
Fusion	Concatenate		
Training	Unimodal, LF	Loss	Cross Entropy
		Batch size	16
		Num epochs	15
		Optimizer	Adam
		Learning rate	0.0001

Table 13: Results on multimodal datasets in the affective computing domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures. MULT is the best performing model on all these datasets, and is categorized as an in-domain method since it was originally proposed and tested on affect recognition datasets. Many out-domain methods struggle on these datasets.

	Dataset Metric	MUSTARD Acc(2) ↑	CMU-MOSI Acc(2) ↑	UR-FUNNY Acc(2) ↑	CMU-MOSEI Acc(2) ↑
U	Unimodal (ℓ)	68.6 ± 0.4	74.2 ± 0.5	58.3 ± 0.2	78.8 ± 1.5
	Unimodal (a)	64.9 ± 0.4	65.5 ± 0.2	57.2 ± 0.9	66.4 ± 0.7
	Unimodal (v)	65.7 ± 0.7	66.3 ± 0.3	57.3 ± 0.5	67.2 ± 0.4
M	EF-GRU	66.3 ± 0.3	73.2 ± 2.2	60.2 ± 0.5	78.4 ± 0.6
	LF-GRU	66.1 ± 0.9	75.2 ± 0.8	62.5 ± 0.5	79.2 ± 0.4
	EF-TRANSFORMER	65.3 ± 1.4	78.8 ± 0.4	62.9 ± 0.2	79.6 ± 0.3
	LF-TRANSFORMER	66.1 ± 0.9	79.6 ± 0.4	63.4 ± 0.3	80.6 ± 0.3
	TF [179]	62.1 ± 2.2	74.4 ± 0.2	61.2 ± 0.4	79.4 ± 0.5
	LRTF [106]	65.2 ± 1.5	76.3 ± 0.3	62.7 ± 0.2	79.6 ± 0.6
	MI-MATRIX [77]	61.8 ± 0.3	73.9 ± 0.4	61.9 ± 0.3	76.5 ± 0.4
	MULT [154]	$\mathbf{71.8 \pm 0.3}$	$\mathbf{83.0 \pm 0.1}$	$\mathbf{66.7 \pm 0.3}$	$\mathbf{82.1 \pm 0.5}$
O	MFM [155]	66.3 ± 0.3	78.1 ± 0.9	62.4 ± 1.1	79.4 ± 0.7
	MVAE [168]	64.5 ± 0.4	77.2 ± 0.3	62.0 ± 0.5	79.1 ± 0.2
	MCTN [123]	63.2 ± 1.4	76.9 ± 2.1	63.2 ± 0.8	76.4 ± 0.4
T	GRADBLEND [167]	66.1 ± 0.3	75.5 ± 0.5	62.3 ± 0.3	78.1 ± 0.3

H Experimental Results

In this section, we provide additional experimental results and observations. For all experimental tables, we describe the accuracy metrics using Acc(c) where c is the number of classes. AUPRC stands for the area under the precision-recall curve which is a useful performance metric for imbalanced data in settings where one cares a lot about finding positive examples. MSE stands for mean squared error. We use up and down arrows (↑ and ↓) to indicate metrics where higher is better (Acc, AUPRC) and metrics where lower is better (MSE) respectively.

H.1 Affective Computing

We show the full performance results in Table 13 and complexity results in Table 14. Here we list some observations regarding these results:

1. Language is usually the best performing modality, especially on sentiment and emotion prediction. However, the improvement of language over audio and video on humor prediction and sarcasm prediction is much less. This follows our intuition that while language is primarily useful for sentiment and emotion prediction, audio and visual are strong predictors for humor and sarcasm.
2. The best performing method over these datasets is consistently the Multimodal Transformer (MULT [155]), which was originally tested on predicting sentiment and emotions on the CMU-MOSI and CMU-MOSEI dataset. We find that it is a general method and generalizes to humor and sarcasm prediction as well.
3. However, while it MULT achieves the best performance, it suffers in complexity, taking more than $12\times$ the inference time of unimodal models and $3-4\times$ several simpler early or late fusion multimodal baselines.
4. Some methods that work well on humor, sentiment, and emotion prediction do not generalize to sarcasm detection, such as tensor fusion (TF) and reconstruction-based models (MVAE and MFM). It is not a surprise that this coincides with sarcasm being the least studied task as well. Furthermore, we believe that it is a task with extremely complementary information (e.g., sarcasm is usually displayed via text and video/audio features contradicting each other). We hope that MULTIBENCH can encourage further research in such multimodal tasks since current methods do not generalize to these tasks.
5. Several out-of-domain methods, such as GRADBLEND do not work well. In fact we find that the variance of the GRADBLEND method is quite high and shows strong performance on several datasets but struggles on others.
6. MCTN is designed for robustness and only uses the language modality at test time. While it was shown to work well for relatively easier fusion tasks in predicting sentiment, emotions, and humor [123], we find that it struggles on the more challenging sarcasm prediction task.

Table 14: Complexity results for datasets in the affective computing domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset	MUSTARD					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (ℓ)	43	381	0.12	2347	0.33	0.12
	Unimodal (v)	48	56	0.01	2288	0.24	0.01
	Unimodal (a)	69	288	0.001	2288	0.25	0.001
M	EF-GRU	126	168	0.84	2291	0.34	0.84
	LF-GRU	74	52	1.52	2307	0.40	1.52
	EF-TRANSFORMER	30	601	1.86	2423	0.79	1.86
	LF-TRANSFORMER	42	1868	14.0	2586	1.02	14.0
	TF [179]	46	1370	14.7	2542	1.62	14.7
	LRTF [106]	33	49	0.68	2483	0.50	0.68
	MULT [154]	31	2414	1.93	3345	3.01	1.93
O	MFM [155]	40	2138	4.85	2417	1.48	4.33
	MVAE [168]	33	4645	4.32	2695	2.11	4.05
	MCTN [123]	100	1026	0.19	2359	1.02	0.19
T	GRADBLEND [167]	100	6012	1.95	2406	0.42	1.58

	Dataset	CMU-MOSI					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (ℓ)	30	590	0.17	2347	0.49	0.17
	Unimodal (v)	35	71	0.01	2288	0.36	0.01
	Unimodal (a)	188	346	0.001	2288	0.38	0.001
M	EF-GRU	106	221	1.42	2291	0.44	1.42
	LF-GRU	14	60	1.84	2307	0.58	1.84
	EF-TRANSFORMER	20	635	2.18	2423	1.07	2.18
	LF-TRANSFORMER	33	2011	15.1	2586	2.12	15.1
	TF [179]	35	384	12.2	2867	2.38	12.2
	LRTF [106]	43	172	0.82	2454	0.59	0.82
	MULT [154]	22	2414	2.38	3345	4.30	2.38
O	MFM [155]	31	1692	5.53	2455	1.52	4.98
	MVAE [168]	35	3820	5.31	2564	2.03	4.69
	MCTN [123]	100	1149	0.19	2366	0.98	0.19
T	GRADBLEND [167]	300	18869	3.91	2355	0.59	1.86

	Dataset	UR-FUNNY					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (ℓ)	32	602	1.99	6524	1.82	1.99
	Unimodal (v)	29	70	0.14	6528	1.61	0.14
	Unimodal (a)	40	1039	0.03	6599	1.66	0.03
M	EF-GRU	34	612	3.58	6535	2.51	3.58
	LF-GRU	10	498	2.28	6791	3.25	2.28
	EF-TRANSFORMER	32	2358	4.87	7086	3.81	4.87
	LF-TRANSFORMER	33	6024	34.5	7288	6.75	34.5
	TF [179]	32	2780	21.3	7165	6.35	21.3
	LRTF [106]	25	2057	1.05	6931	3.32	1.05
	MULT [154]	30	8096	5.01	9572	12.1	5.01
O	MFM [155]	30	5123	6.89	6970	10.3	6.23
	MVAE [168]	32	10670	6.59	7038	12.1	6.10
	MCTN [123]	100	10857	0.19	6578	4.39	0.19
T	GRADBLEND [167]	100	19212	4.12	6832	3.42	2.31

	Dataset	CMU-MOSEI					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (ℓ)	23	561	1.80	5830	1.79	1.80
	Unimodal (v)	27	647	0.12	5817	1.46	0.12
	Unimodal (a)	39	910	0.03	5818	1.48	0.03
M	EF-GRU	22	548	3.23	5835	2.01	3.23
	LF-GRU	9	443	2.08	5996	2.55	2.08
	EF-TRANSFORMER	30	1658	4.49	6082	2.88	4.49
	LF-TRANSFORMER	35	5504	31.5	6996	5.65	31.5
	TF [179]	30	2784	22.6	6337	5.89	22.6
	LRTF [106]	22	2057	0.78	6102	2.45	0.78
	MULT [154]	32	6033	4.75	7572	10.1	4.75
O	MFM [155]	33	5340	6.65	6088	9.42	5.97
	MVAE [168]	40	11673	6.21	6782	12.0	5.89
	MCTN [123]	100	12242	0.19	6526	4.84	0.19
T	GRADBLEND [167]	100	18176	3.89	6042	2.63	2.25

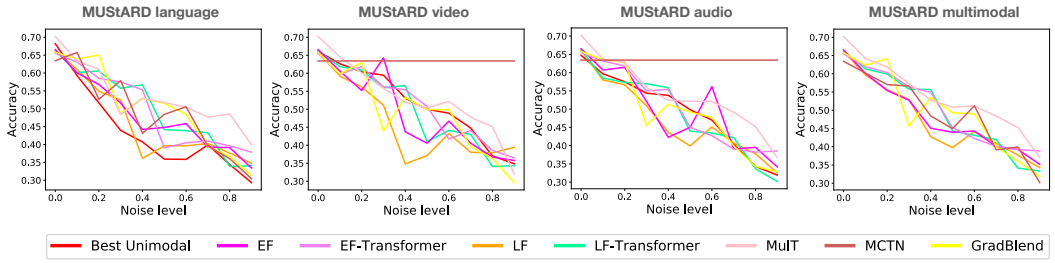


Figure 12: Robustness of multimodal models with increasing levels of noise on the MUSTARD dataset in the affective computing domain.

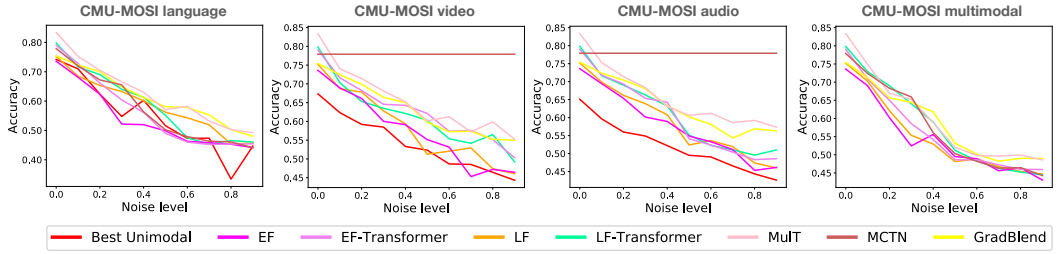


Figure 13: Robustness of multimodal models with increasing levels of noise on the CMU-MOSI dataset in the affective computing domain.

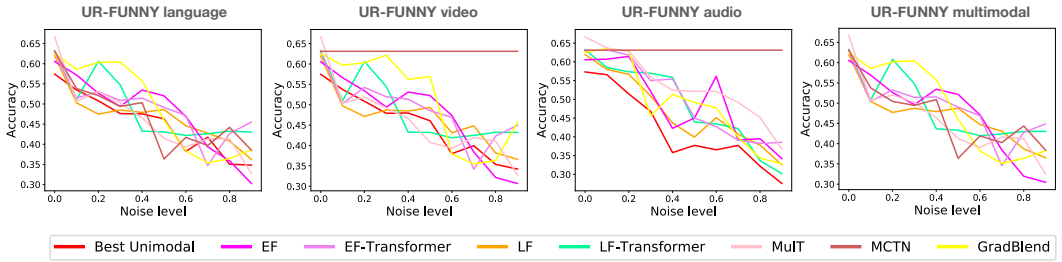


Figure 14: Robustness of multimodal models with increasing levels of noise on the UR-FUNNY dataset in the affective computing domain.

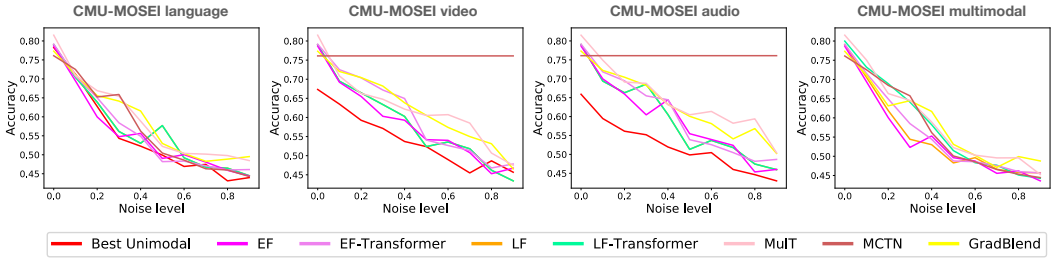


Figure 15: Robustness of multimodal models with increasing levels of noise on the CMU-MOSEI dataset in the affective computing domain.

We show the robustness of multimodal models with increasing levels of noise on MUSTARD in Figure 12, CMU-MOSI in Figure 13, UR-FUNNY in Figure 14, and CMU-MOSEI in Figure 15. We highlight the following observations:

1. Unimodal and multimodal models are in general not robust to increasing noise and imperfections in these datasets. Performance drops off very quickly towards random.
2. We find that multimodal models are slightly more robust than unimodal models. For video and audio, the unimodal method is the least robust. However, for language, the unimodal model can actually be more robust than several multimodal models. In other words, multimodal models are more robust to video and audio while being less robust to language, which is the best performing modality. We believe that directly training multimodal models via supervised learning can be prone to overfitting on the most informative modality (in this case language) which causes the multimodal model to be even less robust than unimodal models in language. A similar observation was the motivation behind the GRADBLEND approach to balance overfitting and generalization across different modalities [167].
3. GRADBLEND [167] seems to be a surprisingly robust approach while also generalizing to several datasets. GRADBLEND was not in fact not initially designed for the affective computing domain, although it was designed for similar multimodal time-series data in the multimedia domain.
4. MCTN [123] was designed as a robust alternative to multimodal models since it uses multimodal data at training time but only language data at test time. On imperfections to video and audio, MCTN therefore stays constant and can potentially be a viable alternative that learns a unimodal model from multimodal data during training but remains unimodal at testing.

Table 15: Results on the MIMIC dataset in the healthcare domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures. Several out-domain methods perform well on MIMIC and improve upon the current state-of-the-art performance on in-domain methods.

	Dataset	MIMIC MORTALITY	MIMIC ICD-9 GROUP 1		MIMIC ICD-9 GROUP 7	
	Metric	Acc(6) ↑	Acc(2) ↑	AUPRC(2) ↑	Acc(2) ↑	AUPRC(2) ↑
	Most frequent	76.1	83.1	–	52.5	–
U	Unimodal (t)	76.7 ± 0.3	83.6 ± 0.1	35.0 ± 0.9	56.3 ± 0.3	54.6 ± 0.4
	Unimodal (ta)	76.4 ± 0.2	91.4 ± 0.0	68.4 ± 0.1	67.6 ± 0.4	72.9 ± 0.3
M	LF	77.9 ± 0.3	91.5 ± 0.1	74.2 ± 0.7	**68.9 ± 0.5**	74.3 ± 0.4
	LRTF [106]	78.2 ± 0.3	91.5 ± 0.1	**75.1 ± 0.3**	68.5 ± 0.4	73.8 ± 0.4
	MI-MATRIX [77]	77.6 ± 0.4	91.5 ± 0.1	74.2 ± 0.6	67.9 ± 0.3	73.0 ± 0.5
	NL GATE [167]	78.1 ± 0.2	**91.6 ± 0.1**	73.8 ± 0.7	68.7 ± 0.5	74.3 ± 0.4
	MFAS [122]	77.9 ± 0.2	91.4 ± 0.0	70.3 ± 1.2	68.5 ± 0.4	73.7 ± 0.4
O	MFM [155]	78.2 ± 0.3	91.5 ± 0.1	75.0 ± 0.5	68.8 ± 0.4	**74.4 ± 0.4**
	MVAE [168]	78.0 ± 0.3	**91.6 ± 0.1**	73.5 ± 1.4	68.7 ± 0.6	74.0 ± 0.7
T	GRADBLEND [167]	**78.2 ± 0.2**	91.5 ± 0.1	74.1 ± 0.4	68.0 ± 0.7	73.2 ± 0.5

Table 16: Complexity results for datasets in the healthcare domain. ((*) This is the number of parameters in modules input to MFAS at the start of training, MFAS will generate more parameters during the architecture search process). **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset	MIMIC					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (t)	20	46.4	0.019	2360	0.41	0.019
	Unimodal (ta)	20	34.6	0.001	2359	0.39	0.001
M	LF	20	49.4	0.034	2362	0.41	0.034
	LRTF [106]	50	261	0.008	2575	0.41	0.008
	MI-MATRIX [77]	20	56.6	0.801	2377	0.39	0.801
	NL GATE [167]	20	51.4	0.040	2422	0.43	0.040
	MFAS [122]	42 × 6	3762	0.086*	2360	1.79	0.016
O	MFM [155]	25	221	0.323	2438	0.85	0.315
	MVAE [168]	30	486	0.312	2553	0.89	0.305
T	GRADBLEND [167]	300	2785	0.063	2575	0.45	0.034

H.2 Healthcare

We show the full results in Table 15 and complexity results in Table 16. Here we list some observations regarding these results:

1. We find that results across all models show small variations on MIMIC, which suggests that many current multimodal approaches may not generalize that well to the input modalities and prediction tasks that MIMIC tests for.
2. In particular, while MFAS (architecture search) is otherwise a pretty general solution that works well across quite a few datasets, it struggles on MIMIC. While there has been a recently proposed MUFASA [173] method that adapts architecture search specifically for healthcare datasets, we were not able to test this method on our partition of MIMIC, and it is in our top priorities to implement that approach into MULTIZOO and accurately benchmark its performance on a suite of datasets.
3. Late Fusion (LF) with simple concatenation was the best-performing model in the evaluations conducted by the previous paper that used the exact same partition as ours [129]. It actually works quite well compared to more complex models evaluated here, as it has the best performance on ICD-9 group 7 task and is quite close to the best performing models in the other two. This may suggest that simple multimodal models such as Late Fusion is worth being tried first on healthcare datasets.
4. The reconstruction-based multimodal models such as MVAE and MFM have strong performance on this dataset, possibly due to the low dimensions of the input modalities. This suggests that reconstruction-based architectures and objectives might work well on datasets with simple or low-dimensional modalities which are easier to reconstruct.

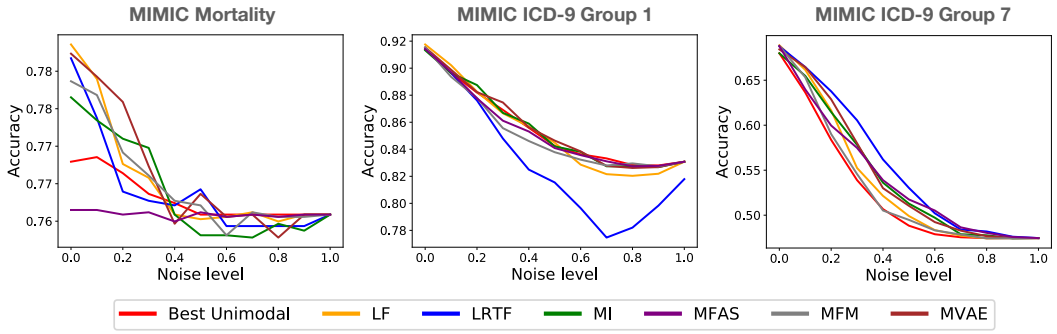


Figure 16: Robustness of multimodal models with increasing levels of noise on the MIMIC dataset in the healthcare domain.

Finally, we show the robustness of multimodal models with increasing levels of noise on the MIMIC dataset in Figure 16. We highlight the following observations:

1. Unimodal and multimodal models are in general not robust to increasing noise and imperfections in the table and time-series modalities. Performance drops off very quickly towards random.
2. In general, multimodal models are slightly more robust than unimodal models. The behavior is best exhibited in the ICD-9 group 7 task where many models start off strong, but multimodal models remain more robust than the best unimodal model. This perhaps indicates that multimodal models do learn to use information from other sources when another one is noisy.
3. There is high variance in the robustness of each multimodal model even within the same dataset and modalities but across different prediction tasks. We observe that LRTF is the most robust model on the ICD-9 group 7 task but the least robust model on the ICD-9 group 1 task. This high variance is a concern especially given the close similarity across both of these tasks.

Table 17: Results on multimodal datasets in the robotics domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset Metric	MuJoCo Push MSE ↓
U	Unimodal (i)	0.334 ± 0.034
	Unimodal (f)	4.266 ± 0.085
	Unimodal (p)	3.885 ± 0.004
	Unimodal (c)	3.804 ± 0.005
M	EF-LSTM	0.363 ± 0.038
	LF-LSTM	$\mathbf{0.290 \pm 0.018}$
	TF [179]	0.574 ± 0.059
	MuLT [156]	0.402 ± 0.026

	Dataset Metric	Vision&Touch Contact Acc(2) ↑	Vision&Touch End Effector MSE ($\times 10^{-4}$) ↓
U	Unimodal (i)	83.6 ± 0.3	1.99 ± 0.160
	Unimodal (f)	$\mathbf{93.6 \pm 0.1}$	87.2 ± 0.477
	Unimodal (p)	85.6 ± 0.6	0.202 ± 0.022
M	LF	$\mathbf{93.6 \pm 0.1}$	$\mathbf{0.185 \pm 0.011}$
	Sensor Fusion [91]	93.4 ± 0.1	0.258 ± 0.011
	LRTF [106]	93.3 ± 0.1	0.232 ± 0.031
O	RefNet [135]	93.5 ± 0.1	0.203 ± 0.025

H.3 Robotics

We show the full results in Table 17 and complexity results in Table 18. Here we list some observations regarding these results:

1. We find that in all robotics tasks, there exists one modality with extremely strong unimodal performance (force in Vision&Touch contact task, proprioception in Vision&Touch End Effector task, image in MuJoCo Push).
2. On the Vision&Touch dataset, we found that Late Fusion outperforms the method of choice in the original paper for the dataset [91] (Sensor Fusion) on both tasks, so Late Fusion seems to generalize well to this domain.
3. In our experiments, as well as the baselines [91], the action modality is typically treated as a general modality without specific modeling. Future work should explore whether this is the best way to encode action as a modality in these action-conditional prediction tasks, and possibly unify these datasets with those used in embodied multimodal learning [36, 97, 110].
4. We plan to include several more reinforcement learning tasks for multimodal learning in robotics. It remains an open question where multimodal representations suitable for fusion-type prediction tasks are also suitable for reinforcement learning tasks. Adding such reinforcement learning tasks from multiple sensors to MultiBench will enable more accurate benchmarking of the generalization capabilities of these multimodal models.

Finally, we show the robustness of multimodal models with increasing levels of noise on MuJoCo Push in Figure 17 and on Vision&Touch in Figure 18. We highlight the following observations:

1. For MuJoCo Push we plot the MSE using a log scale on the y-axis since the error of the TF method blows up significantly much faster than the other methods.
2. We observe that multimodal methods are much more robust than unimodal methods, which match the robustness results as reported in the paper [91] where the trained multimodal model is robust and able to recover from external forces on the force sensor or occlusions to the image sensor. This observation is true for both datasets.
3. For Vision&Touch, we observe that unimodal performance is especially bad for the object pose prediction task. The remaining multimodal models are relatively robust as compared to unimodal performance. The most robust models seem to be Sensor Fusion [91] (SF) and Late Fusion (LF).

Table 18: Complexity results for datasets in the robotics domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset	MuJoCo Push					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (i)	20	738 ± 133	3.88	3607 ± 1	3.46 ± 0.02	3.88
	Unimodal (f)	20	288 ± 39	3.33	3595 ± 2	0.91 ± 0.08	3.33
	Unimodal (p)	20	252 ± 6	3.33	3594 ± 1	0.87 ± 0.04	3.33
	Unimodal (c)	20	372 ± 64	3.33	3594 ± 1	0.86 ± 0.04	3.33
M	EF	20	815 ± 34	3.92	3654 ± 1	4.44 ± 0.55	3.92
	LF-LSTM	20	856 ± 46	1.90	3636 ± 1	4.32 ± 0.45	1.90
	TF-LSTM [179]	20	1914 ± 31	23.5	4530 ± 9	7.75 ± 0.12	23.5
	MuLT [156]	20	4792 ± 62	14.6	6530 ± 16	22.4 ± 0.28	14.6

	Dataset	Vision&Touch					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (i)	15	2633	1.00	5530	63.9	1.00
	Unimodal (f)	15	2185	0.13	2426	51.6	0.13
	Unimodal (p)	15	2514	0.08	2389	59.5	0.08
M	LF	15	2672	1.20	5572	64.4	1.20
	Sensor Fusion [91]	50	11604	1.10	4467	62.6	1.10
	LRTF [106]	35	8366	1.09	4987	64.4	1.09
O	RefNet [135]	15	3819	135	6067	65.0	1.20

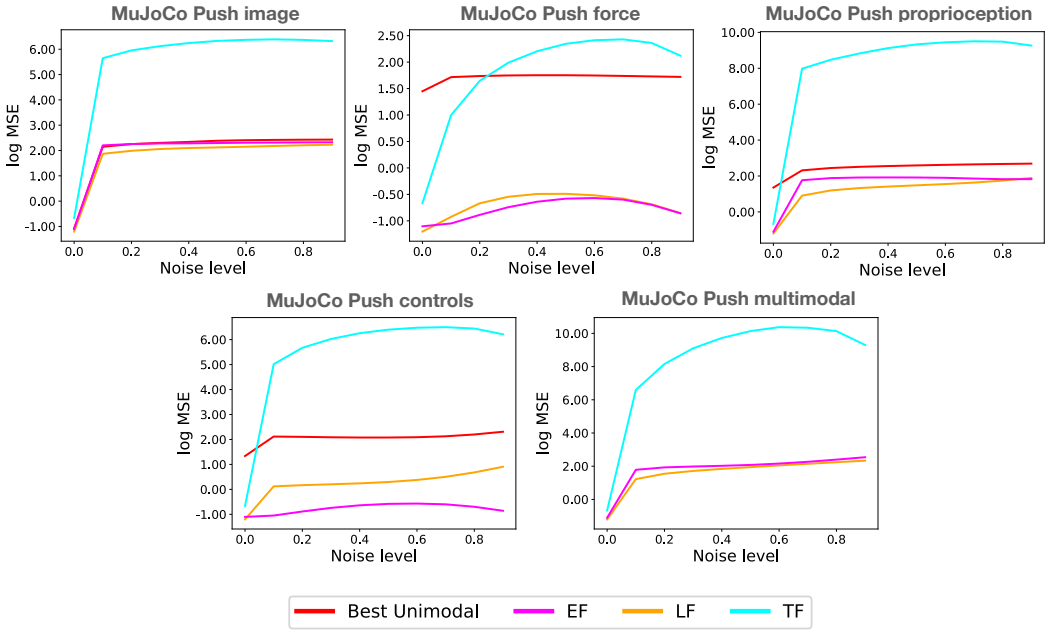


Figure 17: Robustness of multimodal models with increasing levels of noise on the MuJoCo Push dataset in the robotics domain.

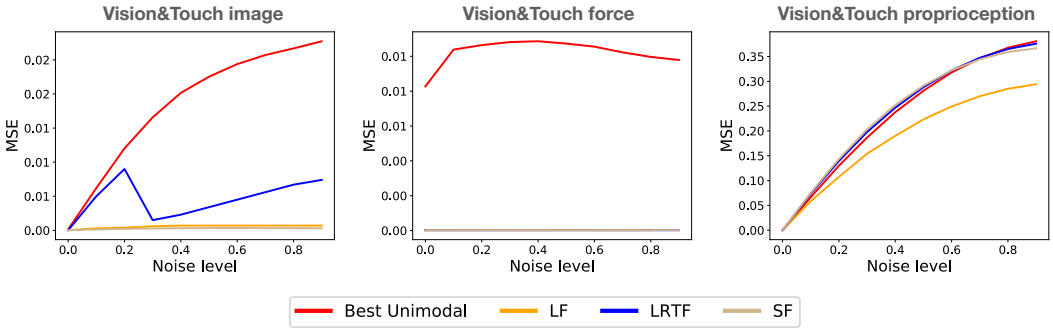


Figure 18: Robustness of multimodal models with increasing levels of noise on the VISION&TOUCH dataset in the robotics domain.

Table 19: Results on multimodal datasets in the finance domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures. MULT struggles on these datasets even though it performs strongly on similar multimodal time-series datasets in the affective computing domain. Other methods also show high variance across different data partitions.

	Dataset Metric	STOCKS-F&B MSE ↓	STOCKS-HEALTH MSE ↓	STOCK-TECH MSE ↓
	Mean	2.140	0.575	0.140
U	ARIMA	2.199	0.620	0.152
	Unimodal	1.856 ± 0.093	0.541 ± 0.010	0.125 ± 0.004
M	EF-LSTM	1.835 ± 0.098	$\mathbf{0.526 \pm 0.017}$	0.121 ± 0.003
	LF-LSTM	1.893 ± 0.106	0.541 ± 0.018	$\mathbf{0.120 \pm 0.008}$
	EF-TRANSFORMER	2.144 ± 0.014	0.573 ± 0.006	0.143 ± 0.003
	LF-TRANSFORMER	2.155 ± 0.023	0.573 ± 0.006	0.143 ± 0.004
	MULT [156]	2.053 ± 0.022	0.555 ± 0.005	0.135 ± 0.003
T	GRADBLEND [167]	$\mathbf{1.820 \pm 0.138}$	0.537 ± 0.011	0.138 ± 0.030

H.4 Finance

We show the full results in Table 19 and complexity results in Table 20. Here we list some observations regarding these results:

1. We do observe better performance using multimodal models as compared to unimodal ones, which suggests that multiple financial signals do help in stock prediction. Several multimodal models do generalize to this more challenging area which presents scalability challenges due to a large number of modalities ($18/63/100$ as compared to $2/3$ in most datasets), as well as robustness challenges arising from real-world data with an inherently low signal-to-noise ratio.
2. There has been very little research in multimodal models in this area, and no public implementations of multimodal models on actual finance data. By adapting current models on this dataset, we observe decent performance of several out of domain methods. Specifically, early fusion (EF) works well which we believe to be due to the little heterogeneity in data origins (i.e., all data comes in the form of time-series data, which is much less heterogeneous as compare to image and text datasets).
3. There remains high variance in the performance of multimodal models even within the same domain: we observe that the best multimodal is not consistent across the 3 partitions of finance datasets, which suggests that current multimodal models remain highly sensitive to the task at hand.
4. Perhaps surprisingly, our experiments on using a Transformer found that they performed worse off than LSTM models. We hypothesize that these large Transformer models might be prone to overfitting on these small and noisy datasets.
5. These datasets present scalability issues to a large number of modalities. We find that we had to adapt several methods such as Tensor Fusion (TF) and Multimodal Transformer (MULT) since they scale exponentially and quadratically with the number of modalities respectively, which does not scale to these finance datasets with more than 10 modalities. We had to adapt these models by performing an initial clustering over the modalities to form $2/3$ groups, performing early fusion by concatenating the data within each group and forming $2/3$ 'modalities' before applying methods such as Tensor Fusion (TF) and Multimodal Transformer (MULT). This might explain their slightly worse performance, especially MULT given its strong performance and generalization to different datasets in the affective computing domain. Future research should focus on more scalable multimodal methods to a large number of modalities. Unfortunately, the bulk of multimodal research being in language and vision means that this question is relatively unexplored.

Finally, we show the robustness of multimodal models with increasing levels of noise on the finance datasets in Figure 19. We highlight the following observations:

1. We again observe a similar trend where the best multimodal models (MULT and sometimes EF) are more robust than the best unimodal model. However, different from other datasets, we find that certain multimodal models can be worse in performance and robustness than the best unimodal model. LF in particular is not very robust and performs worse than the best unimodal method.

Table 20: Complexity results for datasets in the finance domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset	STOCKS-F&B					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (t)	2	9.5 ± 0.1	0.067	3028 ± 3	0.50 ± 0.01	0.067
M	EF-LSTM	2	9.7 ± 0.1	0.069	3067 ± 21	0.51 ± 0.01	0.069
	LF-LSTM	4	62 ± 0.4	0.005	2433 ± 4	1.74 ± 0.02	0.005
	EF-Transformer	4	25 ± 0.3	0.118	2434 ± 3	0.62 ± 0.01	0.118
	LF-Transformer	4	88 ± 0.3	0.472	2468 ± 1	1.70 ± 0.00	0.472
	MulT [156]	4	160 ± 1	0.125	3313 ± 1	4.82 ± 0.06	0.125
T	GRADBLEND [167]	4	409 ± 2	0.338	3102 ± 1	0.44 ± 0.01	0.069

	Dataset	STOCKS-HEALTH					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (t)	2	9.6 ± 0.1	0.067	3032 ± 15	0.51 ± 0.01	0.067
M	EF-LSTM	2	9.6 ± 0.1	0.070	3083 ± 2	0.51 ± 0.02	0.070
	LF-LSTM	4	108 ± 1	0.009	2464 ± 7	2.89 ± 0.04	0.009
	EF-Transformer	4	25 ± 0.4	0.118	2466 ± 4	0.65 ± 0.02	0.118
	LF-Transformer	4	159 ± 1	0.826	2524 ± 1	2.93 ± 0.01	0.826
	MulT [156]	4	162 ± 1	0.125	3315 ± 1	4.88 ± 0.04	0.125
T	GRADBLEND [167]	4	582 ± 4	0.541	3141 ± 2	0.49 ± 0.01	0.070

	Dataset	STOCK-TECH					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (t)	2	9.5 ± 0.1	0.067	3023 ± 1	0.51 ± 0.01	0.067
M	EF-LSTM	2	9.6 ± 0.1	0.070	3075 ± 4	0.53 ± 0.01	0.070
	LF-LSTM	4	92 ± 0.5	0.007	2453 ± 4	2.51 ± 0.04	0.007
	EF-Transformer	4	25 ± 0.4	0.118	2453 ± 1	0.63 ± 0.01	0.118
	LF-Transformer	4	135 ± 1	0.708	2506 ± 1	2.52 ± 0.00	0.708
	MulT [156]	4	161 ± 1	0.125	3315 ± 2	4.79 ± 0.03	0.125
T	GRADBLEND [167]	4	500 ± 3	0.473	3167 ± 1	0.44 ± 0.01	0.070

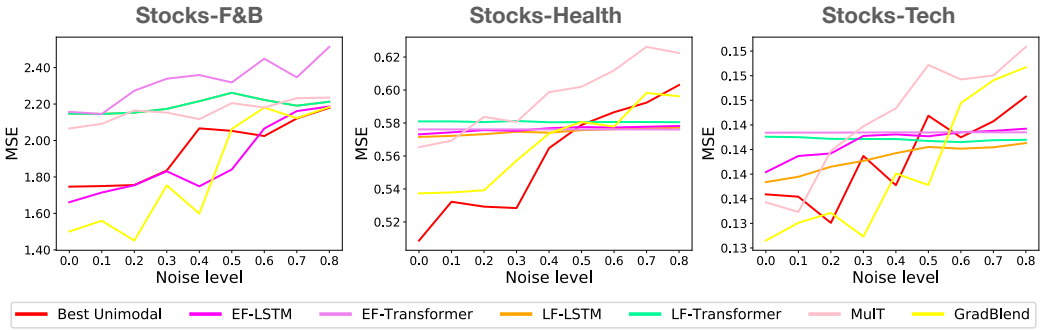


Figure 19: Robustness of multimodal models with increasing levels of noise on the stock prediction datasets in the finance domain.

2. The Gradient Blend (GRADBLEND) method is interesting since it starts off with the best (lowest) MSE but is the least robust – its error increases really quickly and ends up worse than several models that it was initially outperforming on 0 noise levels.
3. We find that several approaches might be underfitting the data on STOCKS-HEALTH and STOCKS-TECH. These methods do not start off with a good MSE and are also not affected significantly at increasing noise levels, showing a roughly straight horizontal line in Figure 19.

Table 21: Results on the ENRICO dataset in the HCI domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures. Several out-domain methods perform well on MIMIC and improve upon the current state-of-the-art performance on in-domain methods.

	Dataset Metric	ENRICO Acc(20) ↑
U	Unimodal (i)	47.0 ± 1.6
	Unimodal (s)	46.1 ± 1.3
M	LF	50.8 ± 2.0
	TF [179]	46.6 ± 1.9
	LRTF [106]	47.1 ± 2.9
	MI-MATRIX [77]	46.7 ± 2.4
O	CCA [145]	50.1 ± 1.4
	REFNET [135]	44.4 ± 2.2
T	GRADBLEND [167]	**51.0 ± 1.4**

Table 22: Complexity results for datasets in the HCI domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset	ENRICO					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (i)	50	1601	9.6	2796	7.3	19.3
	Unimodal (s)	50	1644	9.6	2771	8.1	19.3
M	LF	50	1714	19.3	2730	8.7	19.3
	TF [179]	50	2012	19.3	2718	10.9	19.3
	LRTF [106]	50	1853	19.3	2717	9.7	19.3
	MI-MATRIX [77]	50	1604	19.3	2730	8.5	19.3
O	CCA [145]	50	2945	19.3	2923	9.1	19.3
	REFNET [135]	50	1747	25.7	2757	13.8	25.7
T	GRADBLEND [167]	50	2618	19.3	2610	12.1	19.3

H.5 HCI

We show the full results in Table 21 and results on complexity in Table 22. Here we list some observations regarding these results:

1. The ENRICO paper [93] does not include code or provide many details about their experiments (e.g., data splits, hyperparameters). Compared to their reported results, our reproduction resulted in better performance for the set modality and worse performance for the screenshot modality.
2. Using multiple modalities can help prediction on ENRICO, boosting performance over the best unimodal model by 4%.
3. Similar to finance, there has been very little research in multimodal models for HCI. We observe decent performance of several out of domain methods, especially GRADBLEND which offers a slight improvement over a standard LF model.
4. Certain more complex methods, unfortunately, do not work that well on this dataset. On the architecture side, more expressive methods such as TF, LRTF and MI do not offer improvements over a simple LF model. We hypothesize that these more complex models have a larger number of trainable parameters which make them more prone to overfitting to small and noisy datasets.

We show robustness results with increasing levels of noise in Figure 20. We highlight the following observations:

1. We again observe a similar trend where the best multimodal models (LF and sometimes GRADBLEND) are more robust than the best unimodal model. However, different from other datasets, we find that certain multimodal models can be worse in performance and robustness than the best unimodal model. TF in particular is not robust and performs worse than the best unimodal method.
2. LF is surprisingly robust to imperfections in the image modality and shows a very stable trend despite high levels of noise, implying that the model has learned to rely on the set modality instead when the image is imperfect.

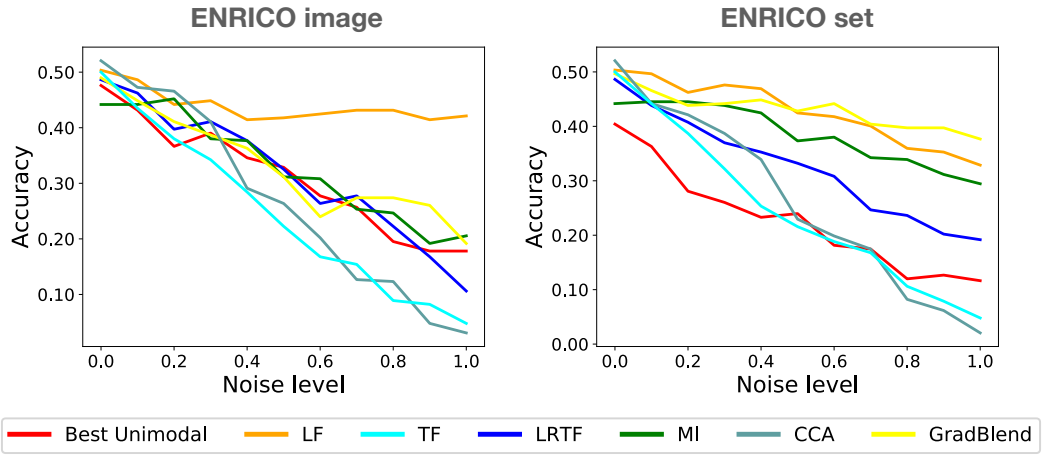


Figure 20: Robustness of multimodal models with increasing levels of noise on the ENRICO dataset in the HCI domain.

3. Multimodal models show a high variance in robustness at high noise levels – performance can range from 5% to 40% at the highest noise levels.

Table 23: Results on multimodal datasets in the multimedia domain. **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures. We observe high variance in model performance across datasets with no method showing consistently strong performance.

	Dataset	MM-IMDB	
	Metric	Micro F1(23) ↑	Macro F1(23) ↑
U	Unimodal (ℓ)	58.6 ± 1.3	45.6 ± 4.5
	Unimodal (i)	40.1 ± 1.3	25.3 ± 0.6
M	EF	58.9 ± 2.6	49.8 ± 1.7
	LF	58.8 ± 1.6	49.2 ± 2.0
	LRTF [106]	59.2 ± 0.5	49.2 ± 0.6
	MI-MATRIX [77]	58.3 ± 1.0	48.0 ± 1.1
O	CCA [145]	**59.3 ± 1.2**	**50.2 ± 0.9**
	REFNET [135]	59.2 ± 2.7	50.2 ± 1.4
	MFM [155]	38.4 ± 1.6	22.3 ± 1.3
T	RMFE	58.6 ± 2.3	47.1 ± 2.0

	Dataset	AV-MNIST
	Metric	Acc(10) ↑
U	Unimodal (i)	65.1 ± 0.2
	Unimodal (a)	42.0 ± 0.2
M	LF	71.7 ± 0.4
	LRTF [106]	71.5 ± 0.5
	MI-MATRIX [77]	71.2 ± 0.5
	MFAS [122]	**72.8 ± 0.2**
O	CCA [145]	71.9 ± 0.4
	REFNET [135]	70.9 ± 0.6
	MFM [155]	71.8 ± 0.4
	MVAE [168]	72.3 ± 0.2
T	GRADBLEND [167]	68.5 ± 0.5

	Dataset	KINETICS-S	KINETICS-L
	Metric	Acc(5) ↑	Acc(400) ↑
U	Unimodal (v)	**56.5**	72.6
	Unimodal (a)	39.7	19.7
M	LF	56.1	71.7
T	GRADBLEND [167]	23.7	**74.7**

H.6 Multimedia

We show the full results in Table 23 and results on complexity in Table 24. Here we list some observations regarding these results:

1. The current SOTA on AV-MNIST is based on architecture search: MFAS [122]. Amongst all the methods we evaluated, MFAS is still the best performing method and beats the second best method (MVAE) by 0.5%. Meanwhile, Gradient Blend (GRADBLEND) does not seem to generalize well to this dataset, as it performs worse than all other multimodal methods.
2. On MM-IMDB, we attempted several methods on the objective function side. We found that using contrastive learning (REFNET) [135] or canonical correlation analysis (CCA) were quite useful in improving performance, with both outperforming purely architectural baselines without alignment as an optimization objective. In particular, while the CCA approach for multimodal fusion was originally proposed for affect recognition datasets [145], we find that they also generalize to the multimedia domain.
3. On KINETICS, Gradient Blend (GRADBLEND) [167] was shown to work really well in their original paper. However, we found that this approach does not generalize well to other datasets such as AV-MNIST. We also created a smaller version of Kinetics called KINETICS-S to enable quick prototyping of multimodal models. Unfortunately, we found that GRADBLEND also struggles on the smaller partition of Kinetics.
4. For KINETICS-S, we also observed that the visual unimodal model slightly outperformed the late fusion model despite the latter using more modalities. This reflects the observations by Wang et al., [167] on the original full version of the KINETICS dataset.

Table 24: Complexity results for datasets in the multimedia domain. ((*) This is the number of params in modules input to MFAS at the start of training, MFAS will generate more params during the architecture search process). **U**: unimodal models, **M**: multimodal fusion paradigms, **O**: optimization objectives, **T**: training structures.

	Dataset	MM-IMDB					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (ℓ)	125	622	0.55	2146	2.07	0.55
	Unimodal (i)	25	127	4.86	2176	2.14	4.86
M	EF	15	117	5.05	2010	3.24	5/05
	LF	5	45	10.3	2016	3.44	10.3
	LRTF [106]	15	741	10.3	2448	5.57	10.3
	MI-MATRIX [77]	20	735	280	4036	3.59	280
	MFM [155]	10	78	21.3	2038	3.36	10.9
	CCA [145]	20	1025	9.51	2273	3.33	9.51
	RMFE [53]	10	104	8.78	22297	3.46	8.78
	REFNET [135]	10	2207	27.0	2899	3.47	10.3

	Dataset	AV-MNIST					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (i)	25	106	0.02	9549	0.95	0.02
	Unimodal (a)	25	158	0.24	11895	1.35	0.24
M	LF	25	260	0.26	11917	1.20	0.26
	MI-MATRIX [77]	25	289	2.53	11509	1.21	2.53
	LRTF [106]	30	470	0.25	11610	1.25	0.25
	MFAS [122]	172×6	17648	0.14^{*}	9444	4.39	0.07
O	CCA [145]	25	310	0.25	9548	1.42	0.25
	REFNET [135]	15	1179	14.01	15931	4.39	0.28
	MFM [155]	25	544	0.92	9570	4.76	0.45
	MVAE [168]	20	679	0.81	9755	4.98	0.34
T	GRADBLEND [167]	300	12539	0.29	12029	1.51	0.26

	Dataset	KINETICS-SMALL					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (v)	15	6702	12.0	12151	13.7	12.0
	Unimodal (a)	15	46767	25.8	8533	60.9	25.8
M	LF	15	20283	37.8	9525	13.9	37.8
T	GRADBLEND [167]	15	20283	37.8	9525	13.9	37.8

	Dataset	KINETICS-LARGE					
	Metric	Epochs trained	Training time (s)	Training params (M)	Training peak memory (MB)	Inference time (s)	Inference params (M)
U	Unimodal (v)	45	938280	12.0	12151	1918	12.0
	Unimodal (a)	45	947380	33.5	8533	8526	33.5
M	LF	45	2839620	45.5	9525	1946	45.5
T	GRADBLEND [167]	45	2839620	45.5	9525	1946	45.5

5. Therefore, we find that multimodal models still struggle on the KINETICS dataset with multimodal performance on simple models (LF) unable to outperform unimodal methods. While GRADBLEND can improve multimodal performance, it comes at the expense of ~ 3× the training time. Future research should explore building lightweight and effective multimodal models on KINETICS as well as other datasets in MULTIBENCH.

We show robustness results with increasing levels of noise on the MM-IMDB datasets in Figure 21. We highlight the following observations:

1. Multimodal models outperform unimodal models when it comes to robustness (and initial performance). This is especially true for imperfections in the image modality. We believe that multimodal models are able to successfully rely on the other modality when one is

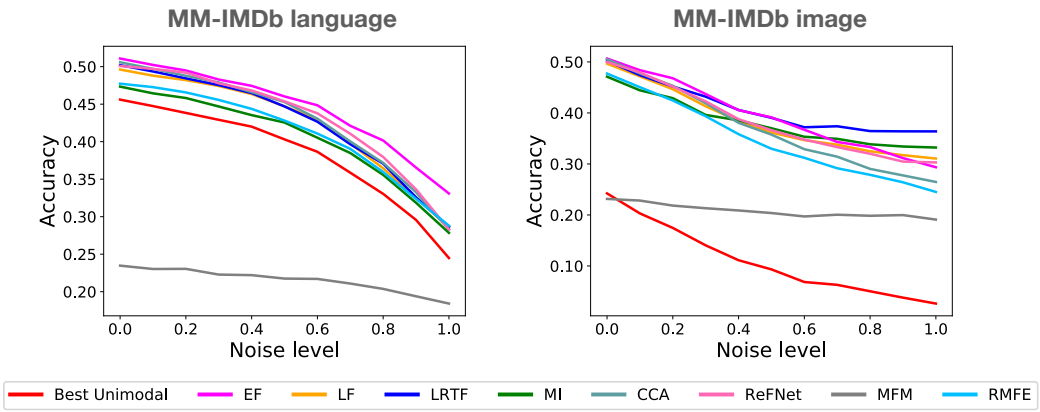


Figure 21: Robustness of multimodal models with increasing levels of noise on the MM-IMDB dataset in the multimedia domain.

imperfect. We find that the gap between multimodal and unimodal performance is very significant on the image modality. However, the gap is much smaller on the text modality.

2. MFM was a method tested initially for affective computing datasets but we found it did not generalize to MM-IMDB, giving poor initial performance and poor robustness. We believe the high-dimensionality of image and text input means that reconstruction of input modalities is difficult, which causes reconstruction-based objectives in MFM to suffer.
3. On the whole, multimodal models are more robust to imperfections on the image modality as compared to the language modality. However, unimodal performance is much better on language than on image, which implies that the language modality is more informative. Similar to the observations on the affective computing datasets, we also find that multimodal models tend to overfit to the more informative modality (language) and are therefore less robust to imperfections in the more informative modality.

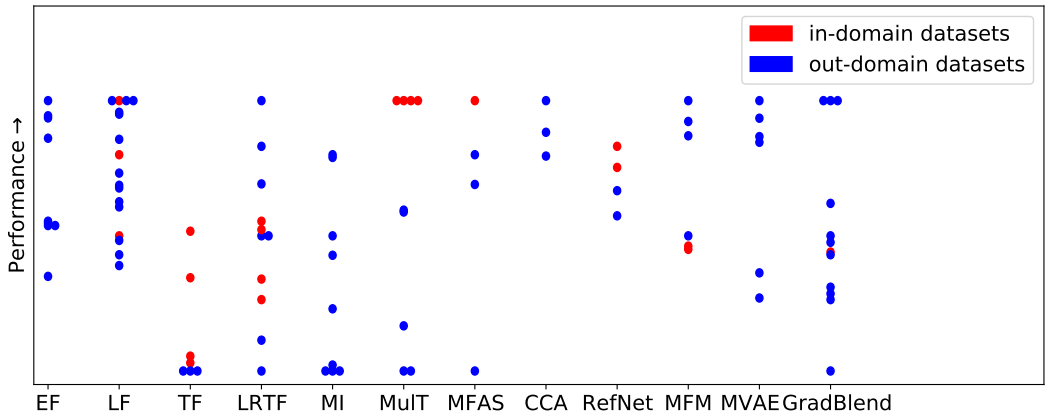


Figure 22: Relative performance of each model across in-domain (red dots) and out-domain datasets (blue dots). *In-domain* refers to the performance on datasets that the method was previously proposed for and *out-domain* shows performance on the remaining datasets. We find that many methods show strongest performance on in-domain datasets which drop when tested on different domains, modalities, and tasks. In general, we also observe high variance in the performance of multimodal methods across datasets in MULTIBENCH, which suggest open questions in building more generalizable models.

H.7 Performance

In this subsection, we summarize several general observations regarding the performance of multimodal models across domains, modalities, and tasks.

In the following analysis, we aggregate the performance of models by first assigning each task a weight of $\frac{1}{n}$ where n is the number of tasks in a dataset (e.g., there are 3 tasks in the MIMIC dataset: mortality, ICD-9 group 1, and ICD-9 group 7 prediction). Then, we compute the scaled performance of each model on each task by min-max normalization – setting the best-performing model's performance to 1 and worst-performing model's performance to 0, and scaling the performance of all remaining models linearly between 0 and 1. Note that we only take the best unimodal performance into account when determining the best and worst-performing models. Then, the final performance score for each model is computed by a weighted average of its scaled performances on all tasks that model was evaluated on.

Benefits of standardization: Simply applying methods in a research different area achieves state-of-the-art performance on 9 out of the 15 tasks. We find that this is especially true for domains and modalities that have been relatively less studied in multimodal research (i.e., healthcare, finance, HCI). Performance gains can be obtained using multimodal methods *outside* of that research area. Therefore, this motivates the benefits of standardizing and unifying areas of research in multimodal machine learning. We believe that the ever-expanding diversity of datasets in MULTIBENCH can greatly accelerate research in multimodal learning.

Generalization across domains and modalities: MULTIBENCH offers an opportunity to analyze algorithmic developments across a large suite of modalities, domains, and tasks. We illustrate these observations through 2 summary plots of the generalization performance of multimodal models. Firstly, in Figure 22, we plot the performance of each multimodal method across all datasets that it is tested on, using the color red to indicate performance on datasets that it was initially proposed and tested on (which we label as *in-domain*), and blue to indicate its performance on the remaining datasets (which we label as *out-domain*). Secondly, in Figure 23, we color-code the performance on each dataset depending on which research area the dataset belongs to (one of the 6 research areas covered in MULTIBENCH).

We summarize several observations regarding generalization across domains and modalities below:

1. Many multimodal methods still do not generalize across domains and datasets. For examples, MFAS [122] works well on domains it was designed for (AV-MNIST and MM-IMDB in the multimedia domain), but does not generalize to other domains such as healthcare (MIMIC). Similarly, the method designed for robustness, MCTN [123], does not generalize to datasets within the affective computing domain (UR-FUNNY and MUSTARD).

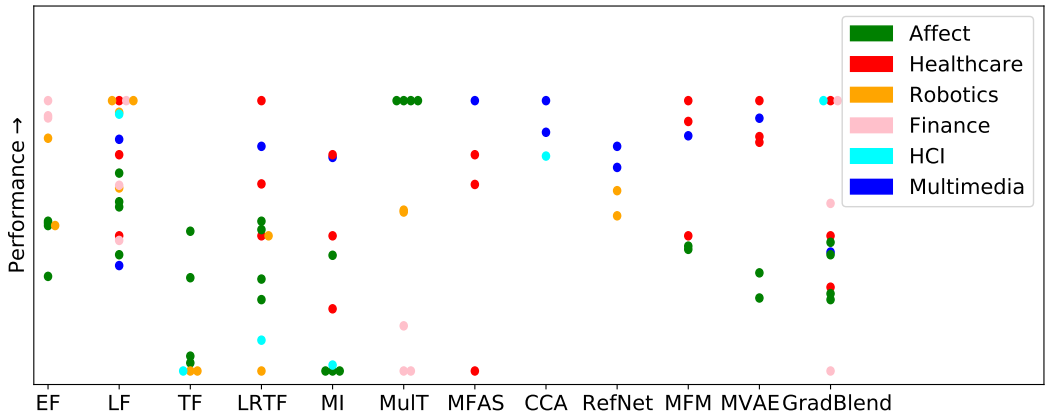
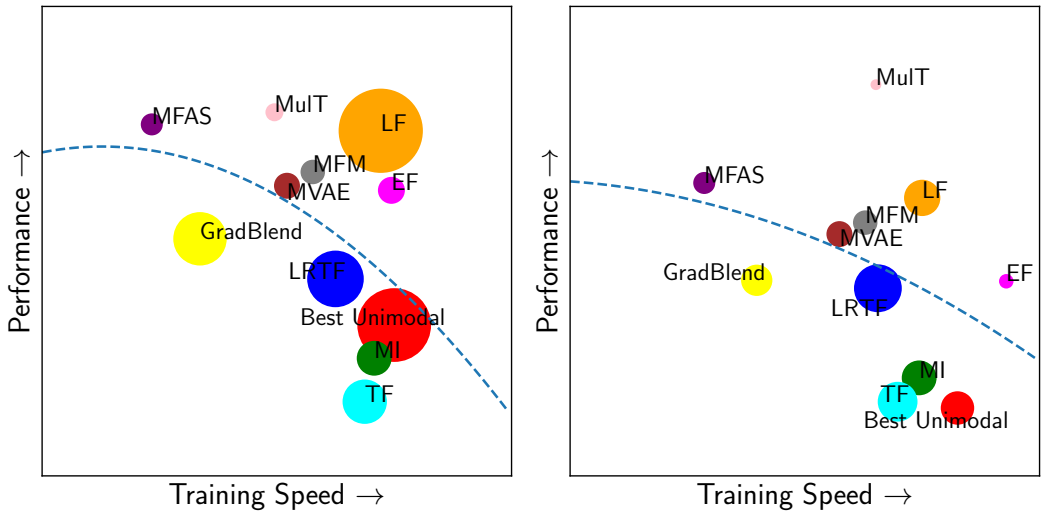


Figure 23: Relative performance of each model across different domains. We find that the performance of multimodal models varies significantly across datasets spanning different research areas and modalities. Similarly, the best-performing methods on each domain are also different. Therefore, there still does not exist a one-size-fits-all model, especially for understudied modalities and tasks.

Finally, GRADBLEND [167], an approach specifically designed to improve generalization in multimodal learning and tested on video and audio datasets (e.g., Kinetics), does not perform well on other datasets. Therefore, there still does not exist a one-size-fits-all model, especially on understudied modalities and tasks.

2. From Figure 22, we find that many methods show their strongest performance on in-domain datasets, and their performance drops when tested on different domains, modalities, and tasks. Some interesting observations are that MULT performs extremely well on the affect recognition datasets it was designed for but struggles on other multimodal time-series in the finance and robotics domains. On the other hand, MFM shows an impressive performance in generalizing to new domains, although its in-domain performance has been exceeded by several other methods.
3. From Figure 22, we also observe high variance in the performance of multimodal methods across datasets in MULTIBENCH, which suggest open questions in building more generalizable models. We find that LF is quite stable and always achieves above-average performance.
4. There are methods that are surprisingly generalizable across datasets. These are typically general modality-agnostic methods such as LF. While simple, it is a strong method that balances simplicity, performance, and low complexity. However, it does not achieve the best performance on any dataset, which suggests that it is a good starting point but perhaps not the best eventual method.
5. From Figure 23, we find that performance also varies significantly across research areas.
6. Several methods such as MFAS and CCA are designed for only 2 modalities (usually image and text), and TF and MI do not scale efficiently beyond 2/3 modalities. Therefore, we were unable to directly adapt these approaches to other datasets. We encourage the community to generalize these approaches across datasets and modalities on MULTIBENCH.

Tradeoffs between modalities: How far can we go with unimodal methods? Surprisingly far! From each of the individual tables, we observe that decent performance can be obtained with the best performing modality. Further improvement via multimodal models may come at the expense of around $2-3\times$ the parameters.



(a) All datasets

(b) Datasets with > 6 approaches

Figure 24: **Tradeoff between performance and complexity**. Size of circles shows variance in performance and complexity across (a) all datasets and (b) datasets on which we tested > 6 approaches. We plot a dotted blue line of best quadratic fit to show the Pareto frontier between performance and complexity. These strong tradeoffs suggest that future work should focus on lightweight multimodal models that generalize throughout datasets in MULTIBENCH. It remains an open question whether several well-performing methods (such as MFAS or MULT) can be successfully adapted to new domains and tasks, since they work much better on more-studied datasets (b) as compared to over all datasets (a).

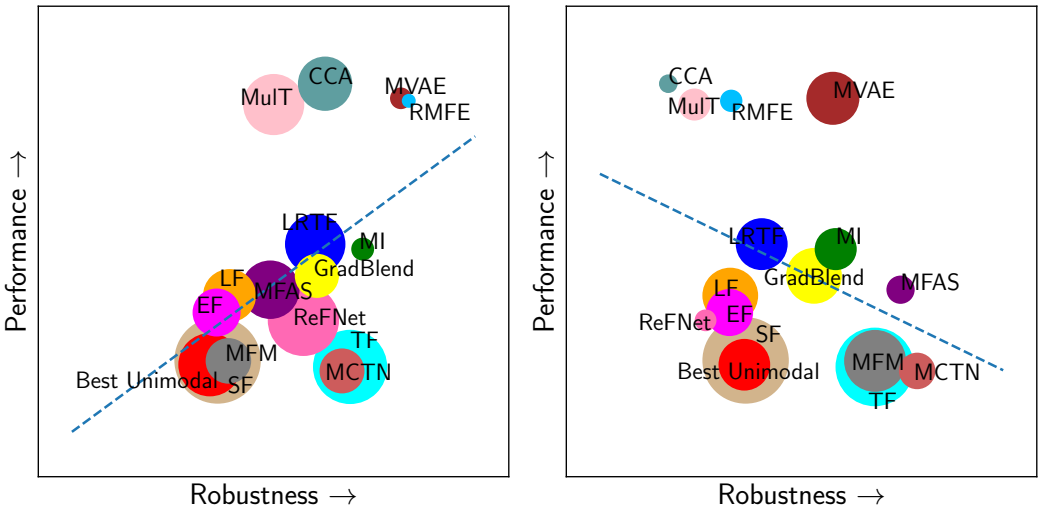
H.8 Complexity

We aggregate the complexity of each model by taking the weighted average of the relative complexity of the model across tasks on which it is evaluated. The weights are assigned in the same way as performance weights described in the subsection above (i.e., performing min-max normalization across models within each task and averaging across the normalized performance across all datasets that the model was tested on). The relative complexity of each model on each task is computed by dividing its training time by the best unimodal model's training time and taking the negative $\log 10$ of this value (we take negative log because some more complex methods can take hundreds of times the training time of simpler methods). Thus, the higher the value of aggregated complexity, the faster the model trains.

Based on the full results above, we summarize the overall tradeoff between performance and complexity in Figure 24(a). We aggregate performance and complexity statistics by first performing min-max normalization within each data to a scale of $0 - 1$ for performance and complexity separately. Note that for metrics where lower is better (i.e., MSE or RMSE) we reverse the direction of min-max normalization. We then aggregate normalized statistics across all datasets and plot the tradeoff between performance and complexity. We highlight the following observations:

1. In Figure 24, we plot a dotted blue line of best quadratic fit to show the Pareto frontier between performance and complexity. We choose a quadratic fit since it is common to fit a curve rather than a straight line when considering the tradeoff frontier between 2 variables (related to the law of diminishing returns in economics). Using this plot, we find a strong tradeoff between these two desiderata: simple fusion techniques (e.g., early fusion EF and late fusion LF) are actually appealing choices that score high on both metrics, especially when compared to complex (but slightly better performing) methods such as architecture search MFAS or Multimodal Transformers MULT.
2. Using this quadratic curve, we find that the best unimodal model is under the curve (i.e., worse-off than the Pareto front). This implies that while unimodal models train the fastest, several multimodal methods can outperform them despite being slightly slower, and is an overall better choice when taking both performance and complexity into account. LF is an appealing choice that lies above the curve.

3. While LF is the easiest to adapt to new datasets and domains, we encountered difficulties in adapting several possibly well-performing methods (such as MFAS or MULT) to new datasets and domains. MFAS is designed with a specific set of atomic architectural elements in mind which makes it most suitable for convolutional networks. MULT is suitable for multimodal time-series data and it is unclear how to adapt its fusion paradigm to modalities without a temporal dimension. For a more fair comparison, in Figure 24(b), we plot the accumulated performance for methods only on the most commonly studied datasets where we experimented with more than 6 methods. We find that these well-performing methods (MFAS or MULT) show only slightly better than LF on all datasets (see Figure 24(a)), they (see Figure 24(b)). Therefore, it is important for future research to focus on models that *generalize* to multiple domains, modalities, and tasks, since our results suggest that many methods currently do not satisfy these desiderata.
4. These plots do not completely capture the picture since complexity is measured via total training time (training speed), which can be prohibitively high for methods such as MFAS, MVAE, and GRADBLEND. However, these methods are primarily slow due to extra parameters or training procedures during training, and once the model is trained, test-time inference is fast and cheap. Plotting a performance-complexity tradeoff using a different complexity metric will likely result in different observations. Overall, MULTIBENCH enables a holistic evaluation of training and test-time space and memory complexity so practitioners can choose the most suitable model for their real-world application setting.



(a) Relative robustness

(b) Effective robustness

Figure 25: **Tradeoff between performance and robustness**. Size of circles shows variance in performance and robustness across datasets. We show the line of best linear fit in dotted blue. While better performing methods display better *relative* robustness (as shown by the positive trend in the best linear fit in (a)), some of these methods suffer in *effective* robustness. In other words, their performance *drops off faster* as shown by the negative linear fit in (b). Few models currently achieve both relative and effective robustness, which suggests a crucial area for future multimodal research.

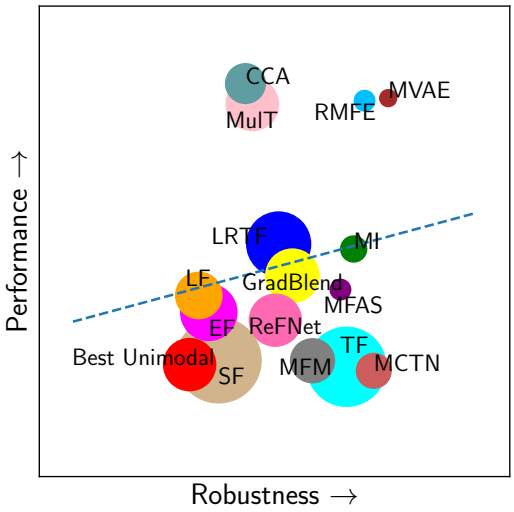


Figure 26: **Overall tradeoff between performance and robustness** obtained by averaging the relative and effective robustness values in Figure 25. We show the line of best linear fit in dotted blue. There is only a slight positive trend between performance and overall robustness of these multimodal models. Therefore, few well-performing models currently achieve both relative and effective robustness, which is a crucial area for future multimodal research.

H.9 Robustness

In this section, we summarize our observations regarding the tradeoffs between accuracy and robustness, where we use the quantitative metrics for relative and effective robustness as described in Appendix D.3. As a reminder, relative robustness directly measures accuracy under imperfections while effective robustness measures the rate of accuracy drops with imperfection after equalizing for initial accuracy on clean test data. In Figure 25, we plot a similar tradeoff plot between accuracy and (relative & effective) robustness. Again, we aggregate performance and complexity statistics by first performing min-max normalization within each data to a scale of $0 - 1$ for performance and

robustness separately. We aggregate normalized statistics across all datasets and plot the tradeoff between performance and robustness. We highlight the following observations:

1. We show the line of best linear fit for relative and effective robustness in dotted blue in Figure 25. We observe a slight positive correlation between performance and relative robustness, which implies that models starting off with higher accuracy tend to stay above other models on the performance-imperfection curve. In particular, several methods such as MVAE and RMFE show strong performance and robustness.
2. However, we observe a slightly negative correlation for effective robustness. Unfortunately, several well-performing methods such as MULT, CCA, and MVAE tend to *drop off faster* after equalizing for initial accuracy on clean test data.
3. Finally, we plot an average of relative and effective robustness in Figure 26 as an overall quantitative measure of robustness. We observe that very few models currently achieve both relative and effective robustness, which prompts an area for future multimodal research.

H.10 Summary of Takeaway Messages

From these results, we emphasize the main take-away messages and motivate several directions for future work:

1. Benefits of standardization: Applying methods in a research different area achieves state-of-the-art performance on 9 out of the 15 datasets, especially those relatively less studied in multimodal research (i.e., healthcare, finance, HCI). This motivates the benefits of standardizing and unifying areas of research in multimodal learning. We hope that MULTIBENCH and MULTIZOO can be a step in this direction.
2. Generalization across domains and modalities:
   (a) Many multimodal methods still do not generalize across domains and datasets, showing high variance across datasets in MULTIBENCH. Some of these methods perform worse on out-of-domain datasets than in-domain datasets while other methods are designed in a specific manner for certain modalities and domains which makes them unable to be adapted to other datasets in straightforward ways.
   (b) Certain simple methods (e.g., LF) are surprisingly generalizable. However, it does not achieve the best performance on any dataset, which suggests that it is a good starting point but perhaps not the best method.
3. Decent performance can be obtained with the best performing modality, which motivates the need for new datasets that offer challenges and opportunities in multimodal modeling not achievable from unimodal methods.
4. There is a strong tradeoff between performance and complexity which suggests that future work should also focus on lightweight multimodal models that generalize throughout datasets in MULTIBENCH.
5. Tradeoffs between performance and robustness:
   (a) Models starting off with higher accuracy tend to stay above other models on the performance-imperfection curve.
   (b) However, several well-performing methods also tend to *drop off faster* after equalizing for initial accuracy on clean test data.
   (c) Overall, very few models currently achieve both relative and effective robustness, which prompts an area for future multimodal research.

I Future Directions

We plan to ensure the continual availability, maintenance, and expansion of MULTIBENCH. Several immediate future directions include expansions in the datasets provided, algorithms implemented in MULTIZOO, and broadening the holistic evaluation of multimodal models.

I.1 Datasets

One main area of expansion lies in the datasets supported by MULTIBENCH. We first describe the categories of multimodal datasets in the fusion domain that we plan to add in the following months. We also plan to include several new application areas where multimodal fusion is useful, such as cross-modal retrieval, multimodal question answering, and grounding across modalities, which we will detail in the following subsections. Finally, we explain our plan for community-based expansion of datasets and models based on user feedback that will happen in parallel.

I.1.1 Fusion

Within the same category of multimodal fusion, we plan to add datasets within the same application domains as well as to expand to new application domains. Within the current domains, we plan to include (1) the hateful memes challenge [82] as a core challenge in multimedia to ensure safer learning from ubiquitous text and images from the internet, (2) more datasets in the robotics and HCI domains where there are many opportunities for multimodal modeling, and (3) several datasets which are of broad interest but are released via licenses that restrict redistribution such as dyadic emotion recognition on IEMOCAP [21], deception prediction on from real-world Trial Data [121], and multilingual affect recognition on CMU-MOSEAS [182] which was only just recently released. We are currently working with the authors to integrate some of these datasets into MULTIBENCH in the near future. These new datasets will benchmark multimodal modeling in human-centric areas where privacy and fairness can be important desiderata. Furthermore, it will enable benchmarking of multimodal learning in languages other than English which is important towards building more accessible multimodal models that include the language modality.

I.1.2 Retrieval

Another area of great interest lies in cross-modal retrieval [104, 187]. In this area, the goal is to retrieve semantically similar data from a new modality using a modality as a query (e.g., given a phrase, retrieve the closest image describing that phrase). The core challenge is to perform alignment of representations across both modalities. Retrieval has been studied primarily in the multimedia space (e.g., retrieving images, video, and audio given a text query) and we hope to add some of these datasets as well as to expand datasets for cross-modal retrieval using different combinations of query and retrieved modalities.

I.1.3 Question Answering

Within the domain of language and vision, there has been growing interest in language-based question answering (i.e., "query" modality) of entities in the visual, video, or embodied domain (i.e., "queried" modality). Datasets such as Visual Question Answering [4], Social IQ [178], and Embodied Question Answering [36] have been proposed to benchmark the performance of multimodal models in these settings. A core challenge lies in aligning words asked in the question with entities in the queried modalities, which can take the form of visual entities in images or videos, and actions in embodied environments. We plan to add these datasets as soon as possible, and also plan to add QA over multiple queried modalities such as text, images, and tables as proposed in recent work [63, 147].

I.1.4 Grounding

Grounding is the task of linking entities (often at their most granular level) in one modality with entities in another modality. As an example, in the domain of language and vision, a well-studied grounding task is visual referring expressions - the task of localizing an object in an image referred to by a natural language expression (e.g., *half of a sandwich on the right side of a plate nearest a coffee mug*) [32]. Grounding can be seen as a more fine-grained version of retrieval where the retrieved modality of interest is at the level of sub-patches of an image. We currently do not include tasks in the grounding area since there are no datasets outside using language to query images (and their subregions). We plan to include grounding datasets in the language and vision domain but also

encourage research in extending this research problem to other modalities (e.g., using language to query video/audio/sets/tables).

I.1.5 Reinforcement Learning

Learning from multiple modalities in an interactive setting is an area of interest as a step towards building more intelligent embodied agents that can perceive the visual world, language instructions, auditory feedback, and other sensor modalities. These research areas broadly span language-conditional RL (i.e., instruction following, learning a reward function from instructions, language in the observation or action space) and language-assisted RL (language as domain knowledge, language to structure policies) [110]. Recent work has also explored audio as a modality in an agent's multisensory interaction with the world [38]. Modern robot systems are also equipped with multiple sensors to aid in their decision-making and there has been considerable research in learning multimodal representations from multiple sensors for robot manipulation [89–91].

These multimodal problems are fundamentally different from those that are concerned with prediction tasks. Alongside the core challenges in learning complementary information and aligning entities in language instructions to those in the visual environment, there also lies the core challenge of learning *actionable* representations that link to the set of actions that can be taken and their associated long-term rewards [110]. We plan to include these datasets in a future version of MULTIBENCH. We also encourage research in extending these multimodal tasks beyond language and vision to truly incorporate the diverse set of modalities humans use in everyday interactive tasks.

I.2 Models

By partitioning the structure of multimodal code into the distinct areas in Appendix E (data processing, unimodal and multimodal model design, optimization objectives, and training structures), MULTIZOO enables easy addition of new innovations from all areas. It is easy to add new unimodal encoders as they are developed in areas such as computer vision and natural language processing. Similarly, it is extremely simple to add multimodal methods while ensuring compatibility with existing unimodal encoders, fusion paradigms, optimization objectives, and training structures. Please refer to Appendix F for code snippets changing multimodal models, optimization objectives, and training structures.

The authors maintain a reading list for topics in multimodal ML [98] that is regularly updated for the latest advances in the area. We plan to periodically add proposed methods to the MULTIZOO toolkit with help from the community as well.

I.3 Evaluation

MULTIBENCH is designed with holistic evaluation in mind. Currently, MULTIBENCH supports evaluation for prediction performance, time and space complexity, and robustness to noisy and missing modalities. There are several other crucial evaluation dimensions that we plan to include in the following versions of the benchmark:

I.3.1 Uncertainty Estimates

There has been important work in building ML models that return uncertainty estimates along with their prediction targets [52, 57] along with recent interest in building multimodal models with similar capabilities [20, 169]. As ML models are increasingly deployed in real-world sensitive scenarios [12, 34, 160], there is an increasing need to quantify when ML models do not know the right answer and potentially abstain [107] or defer the prediction to a human expert [86]. As future steps, we plan to also include evaluations of uncertainty predictions into MULTIBENCH, such as using the recently proposed Uncertainty Toolkit [2, 30, 153]. This will enable the inclusion and evaluation of uncertainty-predicting multimodal models such as the ones proposed in [20, 169].

I.3.2 Robustness to Distribution Shifts

Distribution shifts, spanning shifts in dataset distributions and label distributions, are among core challenges currently preventing machine learning systems from being safely deployed in real-world settings [130]. Subtle changes in the data distribution can significantly impact performance, a phenomenon exemplified by adversarial examples [146], and shifts in the label distribution can significantly compromise accuracy as well [185].

Distribution shifts in multimodal settings have not been explored by the research community. Multimodal data can exhibit shifts in the marginal data distribution of each modality as well as in the joint distribution across modalities, which makes the problem inherently more complex. To enable research in benchmarking and analyzing distribution shift in multimodal settings, we plan to include:

1. *Data:* Data partitions (or new datasets) to MULTIBENCH that test for generalization across domains and subpopulations, in a manner similar to [85]. Building on the current datasets available in MULTIBENCH, some examples include affect recognition across different users, robotic manipulation across different physical robots, and medical diagnosis across different age groups.
2. *Algorithms:* On the algorithmic side, we plan to include currently established methods for distribution shift in a single modality (which has been the bulk of existing work) into MULTIZOO, which will enable both theoreticians and practitioners to analyze the new challenges that multimodal data brings to the study of distribution shift.
3. *Evaluation:* Finally, to evaluate robustness to distribution shift, we plan to build a standardized evaluation pipeline into MULTIBENCH (in a similar way for robustness tests currently implemented). We will also tap into insights from the experimental protocol in [130] which includes evaluation metrics to detect dataset shift before attempting to correct it.

I.3.3 Fairness

To safely deploy human-centric multimodal models in real-world scenarios such as healthcare, HCI, legal systems, and social science, it is necessary to recognize the role they play in shaping social biases and stereotypes. Recent work has shown that word-level embeddings reflect and propagate *social biases* present in training corpora [18, 23]. Machine learning systems that incorporate these word embeddings can further amplify biases [13] and unfairly discriminate against users, particularly those from disadvantaged social groups. Similar observations have been observed for datasets and models in the visual domain such as facial recognition [6] and image captioning [67] tasks, which has called for immediate efforts towards better documentation and risk analysis of both ML datasets [54] and models [115].

We believe that the ability to make fair judgments is even more important in a multimodal setting for the following reasons:

1. Human behavior is inherently multimodal. As a result, many research problems in multimodal learning involve human-centric data and tasks such as healthcare, affective computing, HCI, multimedia, human-robot interaction. As multimodal systems (such as emotion recognition systems) are deployed in the real world, it is crucial to characterize possible social biases they encode and design algorithms to mitigate these biases. Otherwise, real harm can be brought to under-represented populations which unfair machine learning models disproportionately harm [18].
2. While there has been a large body of work investigating the fairness of representations learned from language and images, there is little work currently investigating this for other modalities, as well as for the wide spectrum of multimodal models integrating multiple modalities which can potentially *compound* biases stemming from each one [141].

There are many definitions of fairness and bias in ML and it is unclear which are important in which multimodal settings. While we do not have the best answer to conclusively evaluate for fairness in multimodal systems, we are making it a priority to include this feature in future versions of MULTIBENCH. In reference to [113], certain dimensions of fairness we are currently exploring and plan to add to MULTIBENCH include:

1. *Data:* A better fine-grained understanding of bias in data, which we plan to achieve via human annotations for several multimodal datasets in MULTIBENCH (especially those that involve human-centric tasks such as affect recognition).
2. *Algorithms:* Algorithmic fairness, including training models that satisfy individual and group fairness, analyzing trained models from a geometric perspective (i.e., studying whether biases are encoded in representations learned by a model [18, 99]), and methods for preprocessing and post-processing data and models to satisfy fairness metrics.
3. *Evaluation:* Bias evaluation of trained multimodal models as well as those trained within a single modality, to determine the relationship between biases in a single modality versus

those that manifest in multimodal problems, and comparing them to current progress in this direction on the language and vision modalities [133, 141].

These tasks tackle benchmarking and analysis of biases in multimodal methods from different perspectives spanning data, algorithms, and evaluation, which make them compatible with our proposed modular framework in MULTIBENCH and MULTIZOO. We plan to include additional data annotations in the MULTIBENCH data loader, a suite of algorithms designed to mitigate bias for unimodal and multimodal models in MULTIZOO, and evaluation metrics for fairness in the MULTIBENCH evaluation pipeline.

I.4 Broader Outreach

In workshops and competitions: The authors have extensive experience in organizing challenges, workshops, and tutorials at leading ML, NLP, and computer vision conferences. Among these include large-scale challenges in multimodal language analysis at NAACL 2021 (http://multicomp.cs.cmu.edu/naacl2021multimodalworkshop/), ACL 2020 (http://multicomp.cs.cmu.edu/acl2020multimodalworkshop/), and ACL 2019 (http://multicomp.cs.cmu.edu/acl2018multimodalchallenge/). We plan to use MULTIBENCH as the subject of future workshops to accelerate reproducible research in multimodal learning. These workshops will focus on both new algorithms as well as careful analysis of existing algorithms in the field. Both directions will be accelerated via our resources: we plan to provide MULTIBENCH as a starting point for loading datasets and MULTIZOO as starter code for multimodal modeling, evaluation, and analysis.

In academic courses: We plan to use the MULTIBENCH benchmark as well as the standardized MULTIZOO codebase as an educational tool to support the Multimodal ML course taught annually at CMU (https://cmu-multicomp-lab.github.io/mmml-course/fall2020/). Students can choose to use one of the datasets provided in MULTIBENCH or add a new one to the current suite of multimodal datasets. When designing new algorithmic contributions, students can implement their approaches in the MULTIZOO toolkit which enables easy testing on multiple datasets, quick logging and analysis of results, and reproducible testing. This method of community-based expansion is also likely to see great leaps in the variety of datasets and models supported by this toolkit.

Community-based expansion: Finally, we plan to present a system for expanding the datasets and models in MULTIBENCH via input from the research community. Since MULTIBENCH is publicly released and will be regularly maintained, the existing starting benchmark, code, evaluation, and experimental protocols can greatly accelerate the addition of new datasets and models in the future. In the public GitHub (https://github.com/pliang279/MultiBench), we have included a section on contributing to MULTIBENCH through either task proposals or additions of datasets and algorithms. The readme includes detailed instructions for adding new datasets and dataloaders, as well as new algorithms by modifying according to the code structure we have developed and standardized. The readme also contains details for writing a main function to test new data loaders and multimodal algorithms, and a test script to ensure compatibility with existing experiments. The authors will regularly monitor new proposals through this channel. Periodically, the authors will select popular task proposals (datasets and models) and add it into new versions of MULTIBENCH. The ease of loading datasets and evaluating models will naturally encourage interest in building new datasets and models on top of the toolkit. We further plan to encourage participants/students in our organized workshops and courses to use MULTIBENCH and contribute task proposals as well.