MULTIzoo & MultiBench:
A Standardized Toolkit for Multimodal Deep Learning

Paul Pu Liang, Yiwei Lyu, Xiang Fan, {PLIANG,YLYU1,XIANGFAN}@CS.CMU.EDU
Arav Agarwal, Yun Cheng, {ARAVA,YUNCHENG}@CS.CMU.EDU
Louis-Philippe Morency, Ruslan Salakhutdinov {MORENCY,RSALAKHU}@CS.CMU.EDU

Machine Learning Department and Language Technologies Institute, Carnegie Mellon University

Editor: Antti Honkela

Abstract
Learning multimodal representations involves integrating information from multiple heterogeneous sources of data. In order to accelerate progress towards understudied modalities and tasks while ensuring real-world robustness, we release MULTIzoo, a public toolkit consisting of standardized implementations of > 20 core multimodal algorithms and MultiBench, a large-scale benchmark spanning 15 datasets, 10 modalities, 20 prediction tasks, and 6 research areas. Together, these provide an automated end-to-end machine learning pipeline that simplifies and standardizes data loading, experimental setup, and model evaluation. To enable holistic evaluation, we offer a comprehensive methodology to assess (1) generalization, (2) time and space complexity, and (3) modality robustness. MultiBench paves the way towards a better understanding of the capabilities and limitations of multimodal models, while ensuring ease of use, accessibility, and reproducibility. Our toolkits are publicly available, will be regularly updated, and welcome inputs from the community.

1. MUltiBench was previously published at NeurIPS 2021 (Liang et al., 2021), although the datasets and algorithms were the central contributions of that publication, not the software. This paper focuses on the open-source software along with a larger collection of datasets, algorithms, and evaluation metrics.

©2023 Paul Pu Liang, Yiwei Lyu, Xiang Fan, Arav Agarwal, Yun Cheng, Louis-Philippe Morency, Ruslan Salakhutdinov.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/ Attribution requirements are provided at http://jmlr.org/papers/v24/22-1021.html
Figure 1: MultiBench contains a diverse set of 15 datasets spanning 10 modalities and testing for more than 20 prediction tasks across 6 distinct research areas, and enables standardized, reliable, and reproducible large-scale benchmarking of multimodal models for performance, complexity, and robustness.

Figure 2: Our MultiBench toolkit provides a machine learning pipeline across data processing, data loading, multimodal models, evaluation metrics, and a public leaderboard to encourage accessible, standardized, and reproducible research in multimodal representation learning.

to enable accessibility for new researchers, compositionality of approaches, and reproducibility of results. Together, these public resources ensure ease of use, accessibility, and reproducibility, and they will be continually expanded in courses, workshops, and competitions around the world.

2. MultiBench and MultiZoo

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 2).

MultiBench datasets: Table 1 shows an overview of the datasets provided in MultiBench, which span research areas in multimedia, affective computing, robotics, finance, human-computer interaction, and healthcare, more than 15 datasets, 10 modalities, and 20 prediction tasks.

MultiZoo: A zoo of multimodal algorithms: To complement MultiBench, we release a comprehensive toolkit, MultiZoo, as starter code for multimodal algorithms which implements 20 methods spanning different methodological innovations in (1) data preprocessing, (2) fusion paradigms, (3) optimization objectives, and (4) training procedures (see Figure 3). Each of these algorithms are chosen because they provide unique perspectives to the technical challenges in multimodal learning (Baltrusaitis et al., 2018) (see Table 2 for details).

Evaluation protocol: MultiBench contains evaluation scripts for the following holistic desiderata in multimodal learning: (1) Performance: We standardize evaluation using MSE and MAE for regression, as well as accuracy, micro & macro F1-score, and AUPRC for classification. (2) Complexity: We record the amount of information taken in bits (i.e., data size), the number of model parameters, as well as time and memory resources required during the entire training process. Real-world models may also need to be small and compact to run on mobile devices (Radu et al., 2016) so we also report inference time and memory on CPU and GPU. The datasets and models
Table 1: MULTI-BENCH provides a comprehensive suite of 15 datasets covering a diverse range of 6 research areas, dataset sizes, 10 input modalities (in the form of \( \ell \): language, \( i \): image, \( v \): video, \( a \): audio, \( t \): time-series, \( ta \): tabular, \( f \): force sensor, \( p \): proprioception sensor, \( s \): set, \( o \): optical flow), and 20 prediction tasks.

<table>
<thead>
<tr>
<th>Research Area</th>
<th>Size</th>
<th>Dataset</th>
<th>Modalities</th>
<th># Samples</th>
<th>Prediction task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affective Computing</td>
<td>S</td>
<td>MUSTARD (Castro et al., 2019)</td>
<td>{( i, v, a )}</td>
<td>690</td>
<td>sarcasm</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>CMU-MOSI (Zadeh et al., 2016)</td>
<td>{( i, v, a )}</td>
<td>2,199</td>
<td>sentiment</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>UR-FUNNY (Hasan et al., 2019)</td>
<td>{( i, v, a )}</td>
<td>16,514</td>
<td>humor</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>CMU-MOSEI (Zadeh et al., 2018)</td>
<td>{( i, v, a )}</td>
<td>22,777</td>
<td>sentiment, emotions</td>
</tr>
<tr>
<td>Healthcare</td>
<td>L</td>
<td>MIMIC (Johnson et al., 2016)</td>
<td>{( t, ta )}</td>
<td>36,212</td>
<td>mortality, ICD-9 codes</td>
</tr>
<tr>
<td>Robotics</td>
<td>M</td>
<td>MUJO-CO-PUSH (Lee et al., 2020)</td>
<td>{( i, f, p )}</td>
<td>37,590</td>
<td>object pose</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>VISION&amp;TOUCH (Lee et al., 2019)</td>
<td>{( i, f, p )}</td>
<td>147,000</td>
<td>contact, robot pose</td>
</tr>
<tr>
<td>Finance</td>
<td>M</td>
<td>STOCKS-F&amp;B</td>
<td>{( t \times 18 )}</td>
<td>5,218</td>
<td>stock price, volatility</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>STOCKS-HEALTH</td>
<td>{( t \times 63 )}</td>
<td>5,218</td>
<td>stock price, volatility</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>STOCKS-TECH</td>
<td>{( t \times 100 )}</td>
<td>5,218</td>
<td>stock price, volatility</td>
</tr>
<tr>
<td>HCI</td>
<td>S</td>
<td>ENRICO (Leiva et al., 2020)</td>
<td>{( i, s )}</td>
<td>1,460</td>
<td>design interface</td>
</tr>
<tr>
<td>Multimedia</td>
<td>M</td>
<td>MM-IMDB (Arevalo et al., 2017)</td>
<td>{( i, t )}</td>
<td>25,959</td>
<td>movie genre</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>AV-MNIST (Vielzeuf et al., 2018)</td>
<td>{( i, a )}</td>
<td>70,000</td>
<td>digit</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>KINETICS400 (Kay et al., 2017)</td>
<td>{( v, a, o )}</td>
<td>306,245</td>
<td>human action</td>
</tr>
</tbody>
</table>

Data preprocessing | Unimodal models | Fusion paradigms | Optimization objectives | Training procedures

Figure 3: MULTI-ZOO provides a standardized implementation of multimodal methods in a modular fashion to enable accessibility for new researchers, compositionality of approaches, and reproducibility of results.

Included are designed to span a range of compute times from 1 minute to 6 hours, memory from 2GB to 12GB, models from 0.01 million to 280 million parameters, and datasets from 690 to 147,000 samples. (3) Robustness: The toolkit includes both modality-specific imperfections taking into account each modality’s unique noise topologies (i.e., flips and crops of images, natural misspellings in text, abbreviations in spoken audio), and multimodal imperfections across modalities (e.g., missing modalities, or a chunk of time missing in time-series data) (Liang et al. 2019; Pham et al. 2019).

Installation, testing, and integration: Our documentation provides installation instructions in Linux, MacOS, and Windows. We also include a suite of unit tests (testing self-contained functions) and integration tests (testing multiple components from across the unimodal, fusion, and training loop modules together) with 100% coverage for self-contained functions and 88% coverage overall including integration tests. We also include instructions for continuous integration: our software is hosted on GitHub which enables version control and integration via pull requests and merges. We enabled GitHub Actions workflows, which automatically runs the test builds and is triggered every time new changes are incorporated. After making the desired changes and making sure all tests pass, users can create a pull request and the authors will merge these changes into the main branch.

Together: In Algorithm 1 we show a sample code snippet in Python that loads a dataset, defines the unimodal and multimodal architectures, optimization objective, and training procedures, before running the evaluation protocol. Our toolkit is easy to use and trains models in less than 10 lines of code. By standardizing the implementation of each module and disentangling individual modules, optimizations, and training, MULTI-ZOO ensures accessibility and reproducibility of its algorithms.
Table 2: MULTIZOO provides a standardized implementation of the following multimodal methods spanning data processing, fusion paradigms, optimization objectives, and training procedures, which offer complementary perspectives towards tackling multimodal challenges in alignment, complementarity, and robustness.

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Alignment</th>
<th>Complementarity</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>WORDALIGN (Chen et al., 2017)</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Model</td>
<td>EF, LF (Baltrūšaitis et al., 2018)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>TF (Zadeh et al., 2017), LKTF (Liu et al., 2018)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>MI-MATRIX, MI-VECTOR, MI-SCALAR (Jayakumar et al, 2020)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>NL GATE (Wang et al., 2020)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>MULT (Tsai et al., 2019a)</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>MFAS (Pérez-Rúa et al., 2019)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Objective</td>
<td>CCA (Andrew et al., 2014)</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>RefNet (Sankaran et al., 2021)</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>MFM (Tsai et al., 2019b)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>MVAE (Wu and Goodman, 2018)</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>MCTN (Pham et al., 2019)</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Training</td>
<td>GRADBLEND (Wang et al., 2020)</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>RMFE (Gat et al., 2020)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Algorithm 1 PyTorch code integrating MULTIBENCH datasets and MULTIZOO models.

```python
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

# load Multimodal IMDB dataset
traindata, validdata, testdata = get_dataloader('multimodal_imdb')
out_channels = 3
# define ResNet and Transformer unimodal encoders
collectors = [
    ResNet(in_channels=1, out_channels=3, layers=5),
    Transformer(in_channels=1, out_channels=3, layers=3)]
# define 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)
# train using Gradient Blend algorithm
model = train(collectors, fusion, classifier, traindata, validdata,
              epochs=100, optimtype=torch.optim.SGD, lr=0.01, weight_decay=0.0001)
# test
performance, complexity, robustness = test(model, testdata)
```

3. Results

MULTIZOO and MULTIBENCH enable quick experimentation of multimodal algorithms for performance while balancing complexity and robustness. They uncover several shortcomings of current models, including poor generalization to out-of-domain tasks, tradeoffs between performance and efficiency, and lack of robustness to real-world imperfections. Our resources also pave the way toward answering novel research questions in multimodal transfer learning, multi-task learning, co-learning, pre-training, and interpretability. We include these results and discussions in our full paper (Liang et al., 2021) as well as scripts to reproduce these results in MULTIBENCH software.

4. Conclusion

In conclusion, we present MULTIZOO and MULTIBENCH, a large-scale open-source toolkit unifying previously disjoint efforts in multimodal research with a focus on ease of use, accessibility, and reproducibility, thereby enabling a deeper understanding of multimodal models. Through its unprecedented range of research areas, datasets, modalities, tasks, and evaluation metrics, our toolkit paves the way toward building more generalizable, lightweight, and robust multimodal models.
Acknowledgements

This material is based upon work partially supported by Meta, National Science Foundation awards 1722822 and 1750439, and National Institutes of Health awards R01MH125740, R01MH132225, R01MH096951 and R21MH130767. PPL is partially supported by a Facebook PhD Fellowship and a Carnegie Mellon University’s Center for Machine Learning and Health Fellowship. RS is supported in part by ONR N000141812861, ONR N000142312368 and DARPA/AFRL FA87502321015. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF, NIH, Meta, Carnegie Mellon University’s Center for Machine Learning and Health, ONR, DARPA, or AFRL, and no official endorsement should be inferred. We are extremely grateful to the editor and reviewers for their valuable feedback. Finally, we acknowledge NVIDIA’s GPU support.

References


Paul Pu Liang, Zhun Liu, Yao-Hung Hubert Tsai, Qibin Zhao, Ruslan Salakhutdinov, and Louis-Philippe Morency. Learning representations from imperfect time series data via tensor rank regularization. In ACL, 2019.


