MMMoE: Advancing Multimodal Mixture-of-Experts Architecture

To Power Next-Generation Multimodal Paradigm

Jianzhu Yao, Jul 5th

Dept. Computer Science and Technology, Tsinghua University

Outline

1. Introduction
2. Methods
3. Experiment and Analysis
4. Conclusion and Future Work
5. Reference

Introduction

Motivation

- MoE enlarges the model capacity with more experts but keeps the low computing latency.
- The field of multimodal MoE is under-explored with much potential
- MoE's architecture makes it more suitable for multimodal machine learning such as low-resource scenarios, few-shot learning, or noisy datasets.

Introduction

Contribution

- We present ***MMMoE***, the first multitask multimodal mixture of expert model.
- We propose a heuristic expert selection technique and a modality-level entropy regularisation loss function.
- We analyze the performance, load balance problem, computing latency and few-shot learning scenarios of our model.

Methods

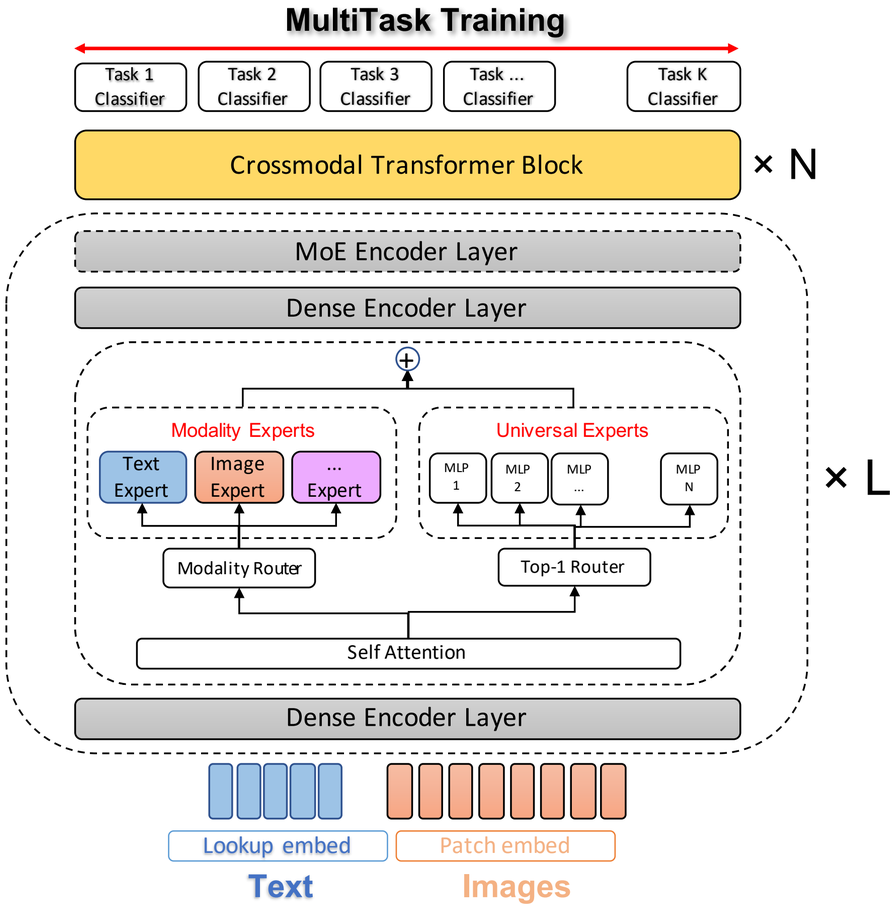
Modality-specific embedding

For each modality $m \in M$, we use a modality-specific embedding network to do the projection.

Just like

lookup embedding for NLP

patch embedding for images



Methods

Heuristic Expert Selection MoE Encoder

Modality expert:

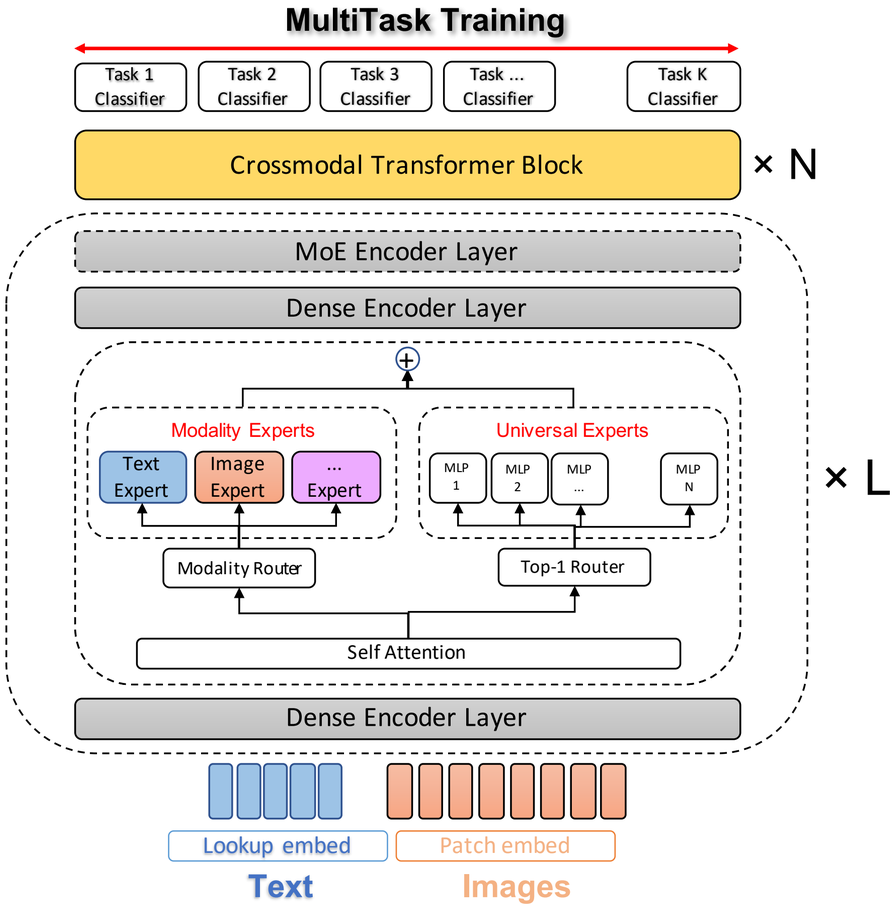
For each token of modality m_i, we use a modality expert E_{m_i} to process.

Universal expert:

For each token, choose E_k with Top-1 routing algorithm of weight g_k.

Final output of this hidden state:

$$h_j^{l+1} = E_{m_i}(h_j^l) + g_k E_k(h_j^l)$$



Methods

Cross-modal Transformer Block

With two modality representations z_1 and z_2, we use a general-purpose Cross-modal Transformer Block with attention:

$ATTN(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$, we set

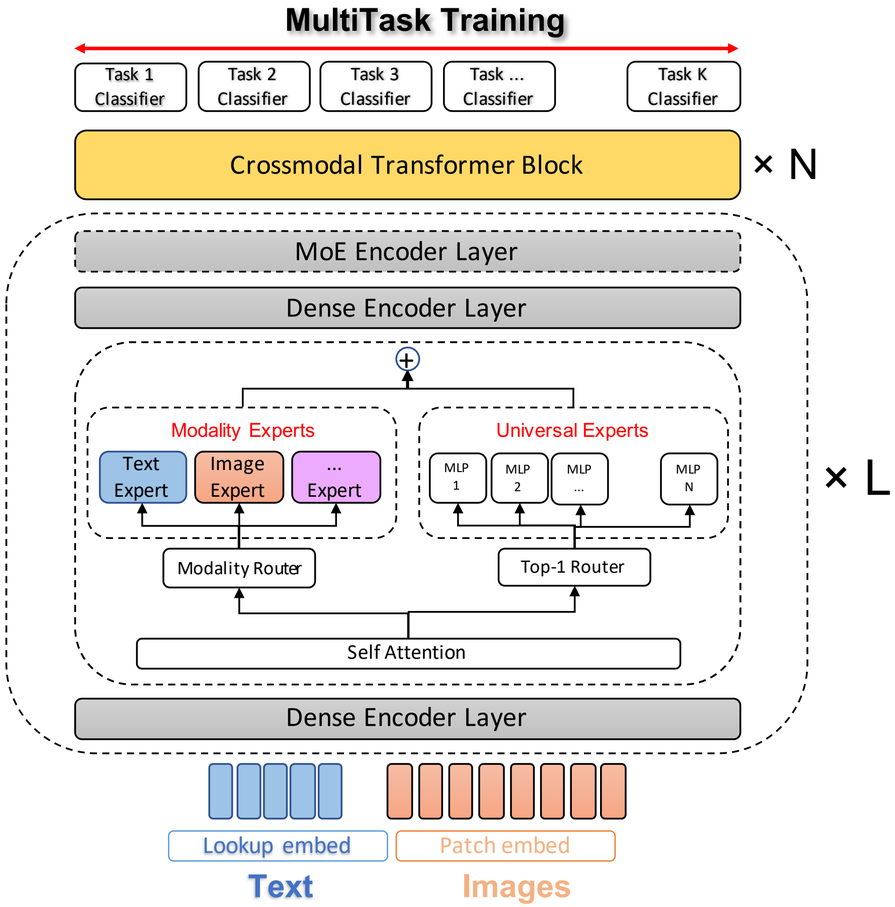
$Q = z_1, K, V = z_2$, to get Z_{12}

$Q = z_2, K, V = z_1$, to get Z_{21}

The final output is

$Z_m = \boldsymbol{W} f([Z_{12}; Z_{21}]) + \boldsymbol{b}$

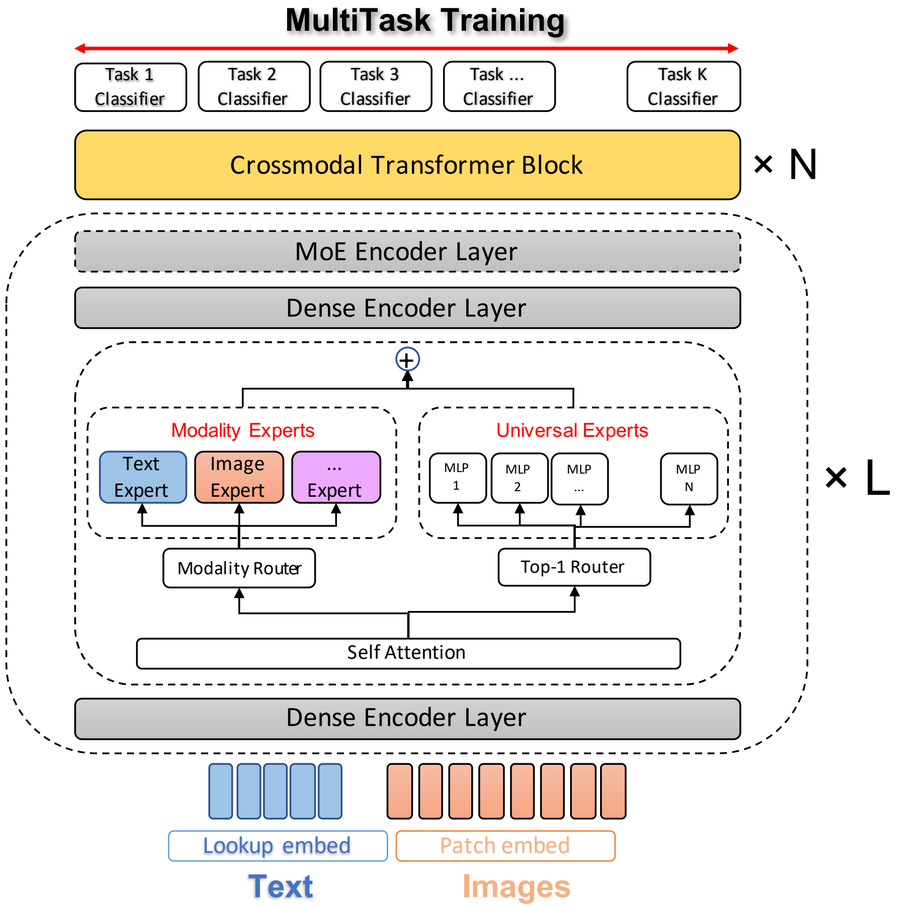
where f is the SiLU activation function.



Methods

Task-specific Classifier and Model Training

For task-specific prediction, we use a linear classification MLP layer per task applied on Z_m to get the final prediction



Methods

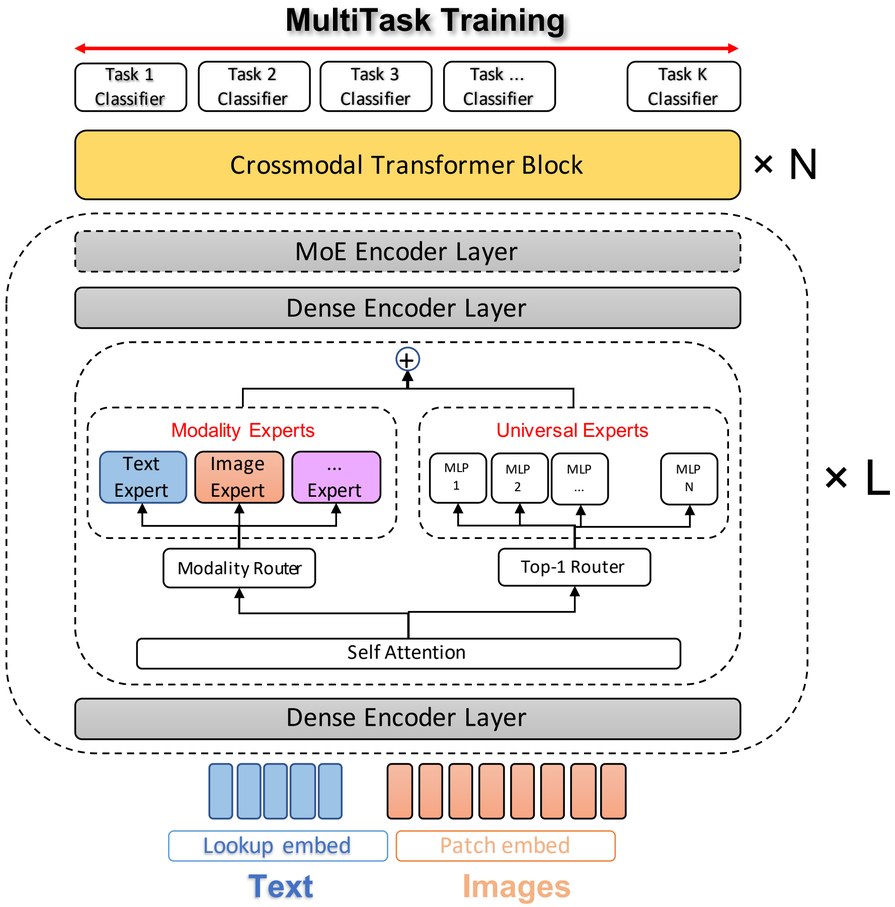
Modality-level Entropy Auxiliary Loss Function

Task loss function: $\mathscr{L}_{Tasks} = \sum_{t=1}^{|T|} w_t \mathscr{L}_t$

Modality-level loss function: $\mathscr{L}^t_{modal} = -\frac{1}{t_m} \sum_{k=1}^{t_m} \mathscr{H}(\tilde{p}_{m_k}(\text{experts}))$

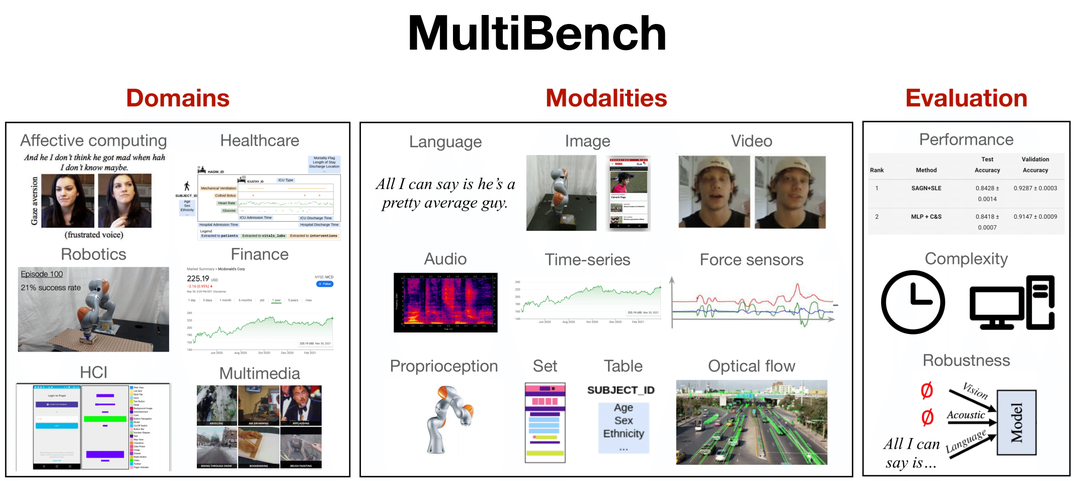
where $\tilde{p}_{m_k}(\text{experts}) = \frac{1}{n_{m_k}} \sum_{i=1}^{n_{m_k}} p_{m_k}(\text{experts} \mid x_i)$, x_i is the i-th token, n_{m_k} is the total token number, and m_k is the k-th modality.

Final loss function: $\mathscr{L} = \mathscr{L}_{Tasks} + \sum_{t=1}^{|T|} w'_t \mathscr{L}^t_{modal}$



Experiment and Analysis

Baseline and Dataset



Dataset: MultiBench

Baseline: (with the same amount of activated parameters)

1. SOTA model provided by MultiBench for each task
2. HIGHMMT, a strong multitask multimodal baseline
3. MoE with Top-2 routing algorithm

(Baseline 1, 2 for performance comparison, 3 for fine-grained experiment analysis of our model.)

Experiment and Analysis

Load Balance Problem — combination of different modalities

To compare the effect of our modality-level loss function, we conduct an experiment on MoE with $\mathscr{L}_{modal} = -\frac{1}{t_m}\sum_{k=1}^{t_m}\mathscr{H}(\tilde{p}_{m_k}(\text{experts}))$ and $\mathscr{L}_{load}(X) = w_{load} \cdot CV(Load(X))^2$

We will visualize and count the number and types of tokens processed by each MoE's layer expert.

Additional Experiments:

Replace $\mathscr{L}_{modal}$ with $\Omega_{local}(\boldsymbol{G}_m)$ and $\Omega_{global}(\boldsymbol{G}_m)$ used in LIMoE.

$$\Omega_{\text{local}}(\mathbf{G}_m) := \frac{1}{n_m}\sum_{i=1}^{n_m}\mathcal{H}(p_m(\texttt{experts}|\mathbf{x}_i))$$

$$\Omega_{\text{global}}(\mathbf{G}_m) := -\mathcal{H}(\tilde{p}_m(\texttt{experts}))$$

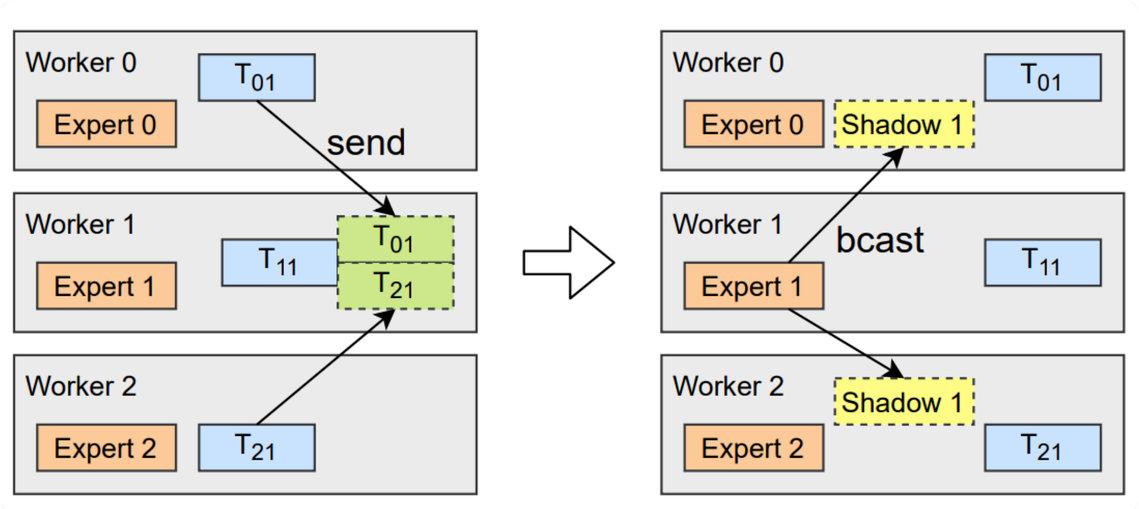
Experiment and Analysis

Computing Latency — Model Profiling

Empirical research is based on observation and measurement of phenomena.

We will conduct a model profiling experiment to do a fine-grained analysis of computing latency, to better lower the communication cost.

- Communication cost of all2all
- The mixture of experts pipeline
- Data and model parallel



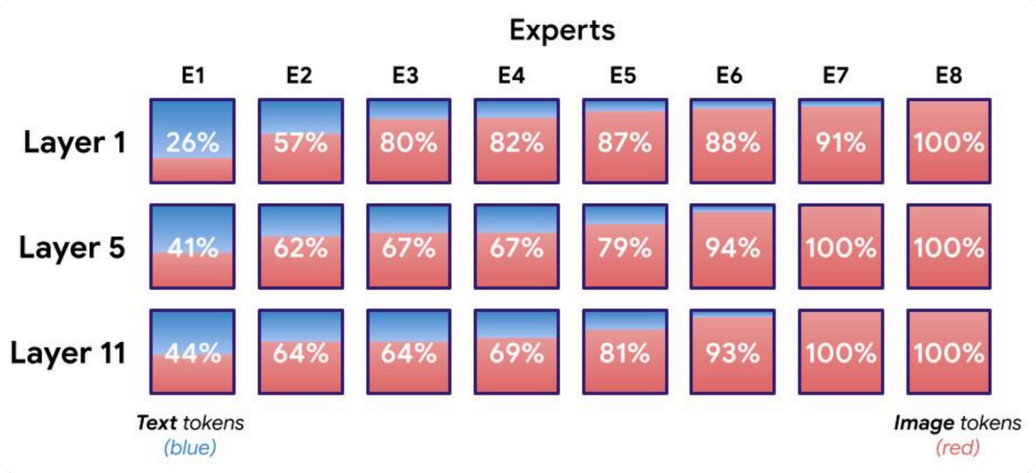
Dynamic shadow the modality experts to each worker

Experiment and Analysis

Few-shot learning experiment

Use two MMMoEs trained on one single task and multitask of MultiBench to compare their performance.

TOGETHER with token distribution analysis, we can visualize the transfer learning details and show how to use cross-modality knowledge to help in such low-resource scenarios.



Conclusion and Future Work

Conclusion

- The new modality-level entropy loss function can handle the load balance problem of multimodal MoEs.
- The heuristic expert selection technique can double the model capacity with the same amount of communication volume as the Top1-gating function.
- The sparsely-activated model performs well in low-resource and noisy scenarios, which is an open problem in the multimodal domain.
- Our approach can reduce computing latency and is environmentally friendly for real-time computing and training.

Conclusion and Future Work

Future Work — Unimodal Sparsely-activated Models

Motivation: The unimodal MoE of different modalities and amazing interactions between each other are under-explored.

Possible Research Path:

1. Conduct unimodal MoE experiments on numerous modalities.
2. Merge to one universal MoE to observe token distribution or other phenomenons.
3. Study on expert pruning or optimized routing algorithm for faster edge computing.

Conclusion and Future Work

Future Work — Multimodal Translation (For Multiple pairs)

MNMT: cross-lingual; Us: cross-MODAL!

Motivation: Such MoE has achieved surprising performance in multilingual translation with transfer learning features and promising expert capacity.

For Multimodal: The shared parameters and collaborated experts will also perform well in low-resource multimodal scenarios.

Reference

1. Multimodal contrastive learning with limoe: the language-image mixture of experts
2. LIMoE: Learning Multiple Modalities with One Sparse Mixture-of-Experts Model (Google AI Blog)
3. Gshard: Scaling giant models with conditional computation and automatic sharding
4. Multimodal transformer for unaligned multimodal language sequences
5. Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks
6. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning
7. Multibench: Multiscale benchmarks for multimodal representation learning
8. Highmmt: Towards modality and task generalization for high-modality representation learning
9. FasterMoE: Modeling and Optimizing Training of Large-Scale Dynamic Pre-Trained Models
10. MultiBench dataset: https://github.com/pliang279/MultiBench