MODALITY LAZINESS: EVERYBODY'S BUSINESS IS NOBODY'S BUSINESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Models fusing multiple modalities receive more information and can outperform their uni-modal counterparts. However, existing multi-modal training approaches often suffer from learning insufficient representations of each modality. We theoretically analyze this phenomenon and prove that with more modalities, the models quickly saturate and ignore the features that are hard-to-learn but important. We name this problem of multi-modal training, *Modality Laziness*. The solution to this problem depends on a notion called paired feature. If there exist no paired features in the data, one may simply run independent training on each modality. Otherwise, we propose Uni-Modal Teacher (UMT), which distills the pre-trained uni-modal features to the corresponding parts in multi-modal models, as a pushing force to tackle the laziness problem. We empirically verify that we can achieve competitive performance on various multi-modal datasets in light of this dichotomy.

1 INTRODUCTION

Multi-modal signals, *e.g.*, vision, sound, text, are ubiquitous in our daily life, allowing us to perceive the world through multiple sensory systems. Inspired by the crucial role that multi-modalities play in human perception and decision (Smith & Gasser, 2005), substantial efforts have been made to build effective and reliable multi-modal systems in fields like multimedia computing (Aytar et al., 2016; Zhao et al., 2018; Wang et al., 2020; Xiao et al., 2020), representation learning (Arandjelovic & Zisserman, 2017; Owens & Efros, 2018; Radford et al., 2021) and robotics (Chen et al., 2020a).

However, existing multi-modal training methods often suffer from learning insufficient representations of each modality, which we term as *Modality Laziness*. Consider a commonly used late-fusion multi-modal network. Different modalities are encoded by their corresponding encoders, and then modern fusion strategies and methods are applied. We follow a standard protocol in self-supervised learning, linear evaluation (Chen et al., 2020c), to assess the feature extraction ability of the encoders. As shown in Table 1 and Figure 2, all encoders from multi-modal training are worse than their uni-modal counterparts, and the strong baseline method, Gradient Blending (Wang et al., 2020), is also no exception.

What results in Modality Laziness? When exposed to more modalities, the model can see more powerful features from different modalities and quickly saturates (the training error becomes zero). As a result, no modality cares about the features that are hard-to-learn but still important, "everybody's business becomes nobody's business". We theoretically characterize this phenomenon and rigorously prove that existing multi-modal training approaches indeed learn fewer features of each modality than uni-modal training, as shown in Figure 1.

We divide the features of multi-modal data into two categories: 1, *self-standing features*, which can be learned in both uni-modal and multi-modal learning; 2, *paired features*, which can *only* be learned in multi-modal joint learning. The solution to the laziness problem depends on paired features. When the paired features are rare, simply running independent training over uni-modal data and then combining the uni-modal models performs well. Otherwise, we propose Uni-Modal Teacher (UMT), which distills the pre-trained uni-modal features to the corresponding parts in multi-modal models while performing multi-modal joint training, as a pushing force to tackle the laziness problem.

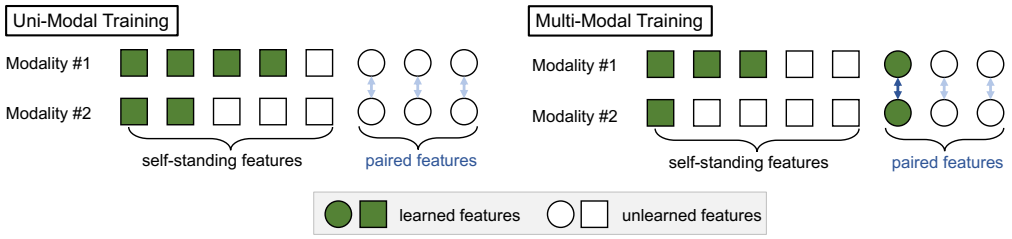


Figure 1: Overview of Modality Laziness. We divide the features of multi-modal data into 1) self-standing features, which can be learned in both uni-modal and multi-modal learning, and 2) paired features, which can *only* be learned in multi-modal joint learning. Although joint training provides the opportunity to learn paired features, multi-modal models are easier to see more powerful features from different modalities and quickly saturate and ignore the features that hard-to-learn but still important. As a result, "everybody's business becomes nobody's business."

In practice, we demonstrate that UMT can effectively improve multi-modal late-fusion learning on datasets like VGG-Sound (Chen et al., 2020b) and UCF101 (Soomro et al., 2012), and it also improves middle-fusion learning in segmentation tasks on NYU Depth V2 dataset (Silberman et al., 2012). We also compare the multi-modal model's performance with that of uni-modal model at the class level, aiming to measure the importance of paired features in different multi-modal datasets (see Table 6). We demonstrate that joint training is important for datasets with more paired features (*e.g.*, VGG-Sound); as for datasets that paired features are rare (*e.g.*, UCF101), combining the individual trained uni-modal models can get competitive results (see Table 7).

We summarize our contributions as follows:

- We introduce linear evaluation to multi-modal late-fusion training and identify an optimization problem in existing methods called Modality Laziness, where encoders from multi-modal training suffer from learning insufficient representations of each modality.
- We theoretically characterize Modality Laziness phenomenon from various aspects. We prove that with multi-modal data as inputs, the model is easier to saturate since it can see more powerful features from different modalities and ignores the features that are hard-to-learn but still important.
- We illustrate the essential roles of the paired feature, which is unique to multi-modal data. When the paired features are rare, simply running independent training over uni-modal data and then combining the uni-modal models performs well. Otherwise, we propose Uni-Modal Teacher (UMT), which employs modality-wise distillation as a pushing force to tackle the laziness problem. And we empirically demonstrate the effectiveness of this dichotomy on various multi-modal datasets.

2 Related Work

Multi-modal training approaches aim to train a multi-modal model by using all modalities simultaneously (Park et al., 2017; Zhao et al., 2018; Hu et al., 2019; Wang et al., 2020; Seichter et al., 2020). While sometimes naively training a multi-modal model even cannot outperform the uni-modal model because of different overfitting rates, Gradient Blending (Wang et al., 2020) introduced adaptive loss weighting to overcome this problem. However, our experimental evidence shows Gradient Blending still suffers from Modality Laziness.

Uni-modal pre-training approaches aim to combine two independent trained uni-modal models (Fayek & Kumar, 2020; Simonyan & Zisserman, 2014; Feichtenhofer et al., 2016). Strong uni-modal models for each modality can be built individually. Although each model can be sufficiently trained on a modality, lacking of joint training would make the approaches fail in multi-modal datasets with many paired features.

Table 1: Top 1 test accuracy (in %) of linear classifiers trained on frozen encoders from various multi-modal late-fusion training methods with uni-modal training on VGG-Sound and UCF101. See §4 for more details.

Method	VGG-Sound		UCF101	
	Video Encoder	Audio Encoder	Flow Encoder	RGB Encoder
Linear-Fusion	15.56	43.44	48.08	75.66
MLP-Fusion	14.52	40.01	51.89	75.65
Attention-Fusion	13.31	43.97	7.72	74.84
Gradient Blending	17.69	43.90	44.49	74.91
Uni-Video (Flow) Training	**23.17**	/	**74.99**	/
Uni-Audio (RGB) Training	/	**45.15**	/	**77.08**

Multi-modal learning theory. The research on multi-modal learning theory is still at an early age. A line of work focuses on understanding multi-view tasks (Amini et al., 2009; Xu et al., 2013; Arora et al., 2016; Allen-Zhu & Li, 2020), and our assumption on the data structure partially stems from Allen-Zhu & Li (2020). Different from the multi-view approaches, when considering multi-modality, one needs to consider the relationship between modalities (*i.e.*, paired feature) beyond multi-view. A representative work on multi-modal learning theory by Huang et al. (2021) provides a theoretical guarantee that learning with multiple modalities achieves a smaller population risk than only using its subset of modalities. Our paper, instead, focuses on the potential negative aspects of multi-modal training (*i.e.*, Modality Laziness).

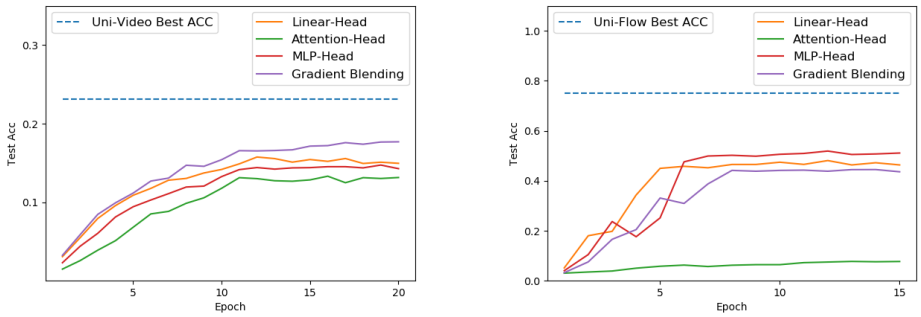
3 Modality Laziness in Multi-Modal Training

3.1 Observations of Modality Laziness

Modality Laziness in multi-modal training. We shall illustrate the existing multi-modal late-fusion training methods suffer from learning insufficient representations of each modality. We call this phenomenon *Modality Laziness*, where the modality with more hard-to-learn features, is significantly under-trained, even when the training of the multi-modal model has already converged. Below we present the experimental evidences in detail (noting that the experimental details can be found in §4.2.):

- As Table 1 shows, video encoder in VGG-Sound from various multi-modal training methods achieves less than 18% accuracy over the testing data in linear evaluation (Chen et al., 2020c), which are significantly lower than 23%, the performance of the video encoder trained over uni-modal data. A similar phenomenon also appears in the flow encoder in UCF101. Even the strong baseline method, Gradient Blending (Wang et al., 2020), cannot avoid insufficient training.
- As Figure 2 shows, throughout the training process, the two selected encoders cannot achieve comparable performance to their uni-modal counterparts. Besides, it is worth noting that the attention head gives more attention to the modality with more easy-to-learn features, which makes the other modality lazier (*e.g.*, video in VGG-Sound and flow in UCF101).
- As shown in Table 2, the depth encoder from ESANet (a multi-modal segmentation method) is also worse than its uni-modal counterpart, which shows that insufficient feature learning phenomenon also exists in multi-modal semantic segmentation tasks with middle-fusion.

To further understand *Modality Laziness* phenomenon, we give the theoretical explanation in the following subsection.



(a) Video encoder evaluation on VGG-Sound. (b) Optical flow encoder evaluation on UCF101.

Figure 2: Linear evaluation on encoders from different multi-modal late-fusion methods. Specifically, we build a linear layer on top of each encoder to receive its detached features. After optimizing the linear layer towards labels, we use the output accuracy of the linear layer as a metric of the corresponding encoder. Here we show the training dynamics of the video encoder in VGG-Sound and the optical flow encoder in UCF101. Other encoder evaluation results can be found in Appendix A.3.

3.2 Theoretical Characterization

The above experiments illustrate that multi-modal training approaches indeed suffer from Modality Laziness issues and learning insufficient representations of each modality. This section characterizes the phenomenon from a theoretical perspective.

Before diving into the technical details, we first provide some intuition behind the proof. Our goal is to show the Modality Laziness issues in multi-modal training approaches, meaning that the model cannot extract sufficient features with limited training samples. We refer to Figure 3 as an illustration[1]. During the training process, learning those easy-to-learn features suffices to reach zero training error (see Figure 3(a), point A). However, the model is not fully trained at point A, and the zero-training-error region (blue) stops us from further training. As a comparison, uni-modal pre-training approaches can break the barrier and achieve point B, which outperforms point A concerning the test error (see Figure 3(b)).

We next prove the modality laziness phenomenon under a simple but effective regime, which characterizes the basic properties of multi-modality learning. We mainly consider cases with two modalities x^{m_1} and x^{m_2}, although similar techniques can be directly generalized to the cases with more modalities.

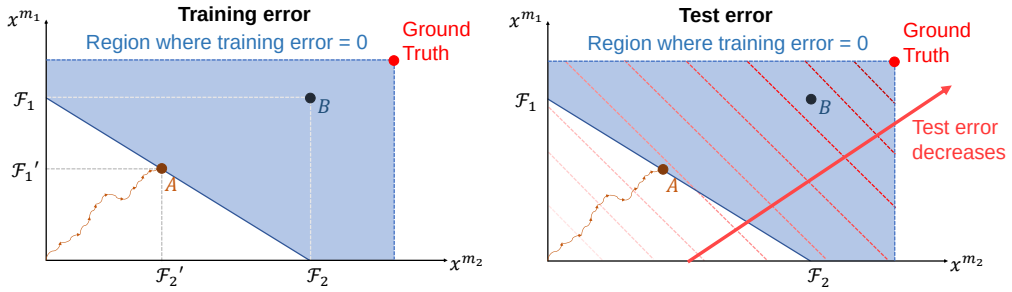
Data distribution. We formalize the data distribution describing how the features are generated. We simplify the data generation process since the multi-modal learning process can be highly complex and hard to characterize. We emphasize that such simplification is still self-contained to describe the differences between self-standing features and paired features.

We highlight that except the self-standing features (defined in Definition 1, learned in both uni-modal approaches and multi-modal training approaches), there exist paired features defined in Definition 2 which can only be learned in multi-modal training approaches. Without loss of generality, consider the binary classification regime where the label y has a uniform distribution over $\{-1, 1\}$. The self-standing features and paired features are generated based on the following two definitions.

Definition 1 (Self-standing features) *The i-th self-standing feature ($f_i(x^{m_1})$) in modality x^{m_1} is generated as:*

$$\begin{aligned} \textit{with probability } p(f_i),\ & yf_i(x^{m_1}) > 0; \\ \textit{with probability } 1 - p(f_i) - \epsilon(f_i),\ & yf_i(x^{m_1}) = 0; \\ \textit{with probability } \epsilon(f_i),\ & yf_i(x^{m_1}) < 0. \end{aligned}$$

[1]We omit the effect of paired features to illustrate the modality laziness phenomenon better.



(a) Training Procedure (b) Testing Procedure

Figure 3: Illustrate for multi-modal training approaches (Point A) and uni-modal pre-training approaches (Point B) under training procedure and test procedure, where the x-axis represents the feature set learned by x^{m_2}, and the y-axis represents the feature set learned by x^{m_1}. The feature set becomes larger along the positive direction of the x-axis, and the training error in the blue region is zero. For uni-modal approaches, x^{m_1} modality learns feature set $\mathcal{F}_1$ while x^{m_2} modality learns feature set $\mathcal{F}_2$ (the intersection between blue region and axis.). In Figure 3(a), multi-modal training approaches learns less features in each modality ($\mathcal{F}_1'$ instead of $\mathcal{F}_1$, $\mathcal{F}_2'$ instead of $\mathcal{F}_2$). In Figure 3(b) where the test error decreases from bottom left to top right, point B (uni-modal pre-training approaches) outperforms point A (multi-modal training approaches).

The i-th self-standing feature ($g_i(x^{m_2})$) in modality x^{m_2} is similarly generated with $p(g_i)$ and $\epsilon(g_i)$.

Definition 2 (Paired features) *The j-th paired feature*[2] h_j *is generated as:*

$$\begin{aligned} \textit{with probability } p(h_j), &\ \ yh_j(x^{m_1})h_j(x^{m_2}) > 0; \\ \textit{with probability } 1-p(h_j)-\epsilon(h_j), &\ \ yh_j(x^{m_1})h_j(x^{m_2}) = 0; \\ \textit{with probability } \epsilon(h_j), &\ \ yh_j(x^{m_1})h_j(x^{m_2}) < 0. \end{aligned}$$

We note that if only one of the paired feature (e.g., $h_j(x^{m_1})$ without $h_j(x^{m_2})$) is used in the model, the predicting ability is relatively low. Therefore, although uni-modal approaches can accidentally learn paired features we do not discuss such rare cases for simplicity.

Remark. The concrete forms in Definition 1 and Definition 2 are not the key points. For example, the XOR form $h_j(x^{m_1})h_j(x^{m_2})$ in Definition 2 can be replaced by any other reasonable terms. However, self-standing features must be generated on one modality while paired features are generated on both modalities.

When the context is clear, we abuse the notation r_i to represent either f_i (self-standing feature in modality x^{m_1}), g_i (self-standing feature in modality x^{m_2}), or h_i (paired feature). We name $p(r_i)$ as the *predicting probability* of feature r_i. When r_i is present (meaning that $r_i \neq 0$), we use $\mathbb{I}(r_i > 0) - \mathbb{I}(r_i < 0)$ to predict y. Otherwise ($r_i = 0$), we random guess y uniformly over $\{-1, 1\}$. To simplify the discussion, we always assume $\epsilon(f_i) = p(f_i)/c$, where $c > 1$ is a fixed constant. For the ease of notations, we define the empty feature in Definition 3.

Definition 3 (Empty Feature) *Empty feature e_i is a kind of self-standing feature (or paired feature) with $p(e_i) = \epsilon(e_i) = 0$.*

Training Procedure. We revisit the two types of training: (a.) *multi-modal training approaches*, which directly train the model using both modality x^{m_1} and modality x^{m_2}; (b.) *uni-modal pre-training approaches*, which first train the features via uni-modal approaches (x^{m_1} and x^{m_2} separately), and then combine the x^{m_1}-learned features and x^{m_2}-learned features. We aim to show that multi-modal training approaches are easier to suffer from insufficient training issues.

During the training process, to simplify the theoretical analysis, we first initialize all the features with empty features e_i. The models then learn the features in descending order of predicting probability,

[2]We abuse the notation h to simplify the notations where $h(x^{m_1})$ and $h(x^{m_2})$ can have different forms.

meaning that the powerful features (with large predicting probability) are learned first[3]. Our goal is to minimize the training error to zero[4].

Evaluation Procedure. We abuse r_i to denote the learned features. For each data point, we random guess $\hat{y}$ on $\{-1, 1\}$ uniformly when $\sum_i \mathbb{I}(r_i > 0) = \sum_i \mathbb{I}(r_i < 0)$. Otherwise, we predict the label by $\hat{y} = 2\mathbb{I}(\sum_i \mathbb{I}(r_i > 0) > \sum_i \mathbb{I}(r_i < 0)) - 1$. We define the error as $\sum_i \mathbb{I}(yr_i < 0) - \sum_i \mathbb{I}(yr_i > 0)$.

Based on the above definitions, we have the following theorem, demonstrating that multi-modal training approaches indeed suffer from insufficient training. Concretely, multi-modal training approaches learn fewer features compared to uni-modal pre-training approaches.

Theorem 1 *Assume that in uni-modal pre-training approaches, the number of features learned in modality x^{m_1} is b_1 and the number of features learned in modality x^{m_2} is b_2. We order the probability of self-standing features (both x^{m_1} and x^{m_2}) in decreasing order of p, namely, $p_{[1]}, \ldots, p_{[i]}$. Assume that multi-modal training approaches learn k_1 self-standing features in modality x^{m_1}, k_2 self-standing features in modality x^{m_2}, and k_3 paired features with predicting probability $p(h_1), \ldots, p(h_{k_3})$. Then the following statements hold:*

- *(a.) **Quantity Laziness**: $k_1 + k_2 + k_3 \leq \min\{b_1, b_2\}$.*
- *(b.) **Uni-modal Laziness**: Each modality in multi-modal training approaches performs worse than uni-modal approaches.*
- *(c.) **Performance Laziness**: Consider a new testing point, for every $\delta > 0$, if $\sum_{i\in[k_3]} p(h_i) \leq \sum_{i\in[b_1+1,b_1+b_2]} p_{[i]} + \sqrt{8(k_3 + b_1 - k_1 + b_2 - k_2)\log(1/\delta)}$, then uni-modal pre-training approaches outperform multi-modal training approaches concerning the loss on the testing point with probability at least $1 - \delta$, where the probability is taken over the randomness of the testing point.*

In theorem 1, we describe the modality laziness in three aspects: **Quantity Laziness** introduces the modality laziness from the feature number perspective, indicating that the number of features learned in multi-modal training approaches is less than any of that in uni-modal approaches. **Uni-modal Laziness** comes from the performance of each modality, showing that multi-modal training approaches in each modality perform worse than any of the modality in uni-modal approaches. **Performance Laziness** compares the performance of multi-modal training approaches and uni-modal pre-training approaches, demonstrating that with rare paired feature, uni-modal pre-training approaches outperforms multi-modal training approaches, resulting from the modality laziness.

The theory meets the experimental results in Section 3.1 perfectly well, indicating that the assumptions and the models used in Theorem 1 indeed characterize the reality. We defer the complete proof to Appendix B.1 due to limited space. Besides, we give a concrete example in Appendix B.3 to better illustrate Theorem 1. We remark that uni-modal pre-training approaches are still not perfect since uni-modal pre-training approaches cannot extract the information of paired features. This inspires us to explore more when there exists numerous paired features in Section 3.3.

3.3 Fighting Against Modality Laziness

Based on the discussion in Section 3.2, to solve the modality laziness issues, we need to categorize the problem according to the paired features. When paired features are rare, uni-modal pre-training approaches can already act as a pushing force (see Figure 2 and Performance laziness in Theorem 1), which helps break the laziness barrier. However, when there exist numerous paired features, uni-modal pre-training approaches cannot fully take advantage of the multi-modality since they cannot learn the paired features explicitly. This forces us to reconsider the training process with multi-modalities and introduce a new pushing force on multi-modal training approaches.

Distillation can act as the pushing force, where we first train the teacher models using uni-modal approaches and then apply multi-modal training approaches with distillation on the teacher mod-

[3]We note that recent works have demonstrated that neural networks indeed prefer easy-to-learn features (Shah et al., 2020; Pezeshki et al., 2020).

[4]We always assume that the training error can be minimized to zero.

Algorithm 1 Uni-Modal Teacher (UMT) Framework

Input: Uni-modal pre-trained models f_1, f_2, initialized multi-modal model f_m, iteration number N, loss weight $\lambda_{task}, \lambda_{distill_1}, \lambda_{distill_2}$
for 0 **to** N **do**
  Sample a batch multi-modal data $\{(X_{m_1}, X_{m_2}, Y\} \sim \mathcal{D}$
  Compute the uni-modal target features $f_1^{target}, f_2^{target}$ from uni-modal pre-trained models
  Compute the uni-modal feature and the prediction from the multi-modal model f_1, f_2, Y_{hat}
  Compute the loss between Y_{hat}, f_1, f_2 and $Y, f_1^{target}, f_2^{target}$ and multiply by the $\lambda_{task}, \lambda_{distill_1}, \lambda_{distill_2}$, respectively.
  Update the multi-modal model by SGD or its variant.
end for
Return: A trained multi-modal model

els (Uni-Modal Teacher, UMT). The advantages of distillation to help overcome modality laziness can be divided into two folds. Firstly, distillation changes the learning priority since models prefer to learn the distilled features. During the analysis, we formulate such changes as a boosting on the surrogate predicting probability, which only changes the training priority but does not change the actual predicting ability (See Example 2 as an example). Secondly, even if the training error is zero, the distillation loss is still non-zero and forces the training process to continue. We provide the algorithm details of UMT in Algorithm 1 and empirically validate the algorithm in Section 4.3. We next prove that UMT outperforms uni-modal pre-training approaches in Theorem 2.

Theorem 2 *Denote the paired features by $h_1, \dots h_L$ with corresponding predicting probability $p(h_1), \dots, p(h_L)$. Assume that distillation can boost the training priority by $p^0 > 0$. If there exists paired feature whose predicting probability exceeds the boosting probability p^0, namely, the set $\mathcal{S}$ is not empty:*

$$\mathcal{S} = \{h_i : p(h_i) > p^0\} \neq \phi.$$

Then UMT can learn paired feature which cannot be learned by uni-modal pre-training approaches, namely, UMT outperforms uni-modal pre-training approaches.

4 Experiments

4.1 Experimental setup

In this subsection, we describe the datasets and the backbone models used.

VGG-Sound (Chen et al., 2020b) is an audio-visual classification dataset which contains over 200k video clips for 309 different sound classes.

UCF101 (Soomro et al., 2012) is an action recognition dataset with 101 action categories, including 7k videos for training and 3k for testing. And we use the rgb and optical flow provided by (Feichtenhofer et al., 2016).

NYU Depth V2 (Silberman et al., 2012) contains 1449 indoor RGB-Depth data in total and we use the 40-class label setting. The number of training set and testing set is 795 and 654 respectively.

Table 2: Depth encoder evaluation on RGB-Depth semantic segmentation setting. "Initialization" indicates how weights are initialized for the network, "Uni-Depth" represents end-to-end training with a depth-only segmentation network, and "from RGB+depth" refers to freezing the depth encoder (ResNet-34) from ESANet (Seichter et al., 2020) then fine-tuning with a new decoder.

Initialization	**Training Setting**	
	Uni-Depth	from ESANet
From Scratch	32.69	*28.53 (-4.16)*
ImageNet Pre-train	39.45	*34.73 (-4.72)*

Backbone architectures. In classification on VGG-Sound and UCF101, we use ResNet as our backbone, all with 18 layers (noting that 3D CNN is used for visual data of VGG-Sound). We experiment with different heads for fusion, including linear head, MLP head and attention head. For semantic segmentation on NYU Depth

V2, we use a U-Net like encoder-decoder architecture based on ESANet (we choose ResNet-34 as the encoder), a state-of-the-art multi-modal segmentation method, and more details can be found in Seichter et al. (2020).

The data preprocessing, training hyper-parameters, optimizer, and other details can be found in the Appendix A.1 and A.2.

4.2 Encoder Evaluation in Multi-modal training

Linear evaluation is a commonly used technique to evaluate the encoder in self-supervised learning (Chen et al., 2020c). Specifically, we train an initialized linear classifier on the trained frozen encoder. By checking the top-1 test accuracy of the classifier, we can know the encoder's feature extraction ability. As Table 1 shows, all encoders from multi-modal training get worse top-1 test accuracy in linear evaluation compared to its corresponding uni-modal counterparts, especially the video encoder in VGG-Sound and the optical flow encoder in UCF101. Besides, we build a linear classifier on each encoder to monitor the encoders' dynamic in the training process. This classifier receives the detached feature from its corresponding encoder and is optimized towards labels in each iteration without affecting the encoder. As shown in Figure 2, throughout the training process, the two selected encoders cannot achieve comparable performance to their uni-modal counterparts.

Evaluate the segmentation encoder. Different from classification, in multi-modal middle-fusion segmentation task, the depth feature maps are fused to the RGB backbone in the middle of the encoders (Seichter et al., 2020), which means we cannot compare the RGB encoder from the multi-modal architecture with the uni-RGB model's. Hence we perform evaluation only on the depth encoder. As shown in Table 2, Modality Laziness also emerges in segmentation. Noting that in segmentation task, we train a new decoder (the same as the decoder used in Seichter et al. (2020)) over the trained encoder for pixel-wise prediction.

4.3 Uni-modal teacher is an effective pushing force in multi-modal training

In this subsection, we compare UMT with other multi-modal training methods. Noting that the implementation details of UMT in multi-modal classification and segmentation can be found in Alg 1 and Appendix A.5.

Table 3: Results of different multi-modal training methods. See §4.3 for details.

Method	VGG-Sound	UCF101
Linear-Head	49.46	82.32
MLP-Head	44.76	79.96
Attension-Head	49.80	74.15
Aux-CELoss	49.86	81.34
G-Blending	50.39	83.03
UMT	**53.46**	**83.71**

Table 4: Results of Self-Distillation and UMT. See §4.3 for more details.

Method	VGG-Sound
Naive Baseline	49.46
Self-Distill (label)	49.67
Self-Distill (feature)	49.86
UMT (Contrastive)	52.10
UMT (L1 loss)	52.80
UMT (MSE loss)	**53.46**

For classification. The late-fusion architecture is used for the classification task. The features are extracted from different modalities by the corresponding encoders and then mapped to the output space by the head layers. Specifically, we use the linear layer, MLP, and attention layer as the head, respectively. Auxiliary-CEloss means adding extra linear heads to receive the corresponding uni-modal features and then generating additional cross entropy losses. Auxiliary-CEloss gives all losses equal weights, while Gradient-Blending reweights the losses mainly according to the overfitting-to-generalization-ratio (OGR) (Wang et al., 2020). As shown in

Table 5: Model performance comparison under UMT and ESANet on NYU-DepthV2 RGB-Depth semantic segmentation task.

Initialization	Training Setting	
	ESANet	UMT
From Scratch	38.59	**40.45 (+1.86)**
ImageNet Pre-train	48.48	**49.39 (+0.91)**

Table 6: Top-1 test accuracy of different models on selected classes of VGG-Sound. It appears that the multi-modal naive fusion model outperforms other uni-modal models in these classes, and even exceeding the sum of the accuracy of the uni-audio model and uni-video model. However, we do not find any classes like those in UCF101, meaning VGG-Sound contains more paired features. More details can be found in Appendix A.10

Class ID	164	303	33	255	91	4	152	127	68	155
Uni-Audio	30%	7%	34%	10%	43%	50%	18%	0	53%	32%
Uni-Video	3%	2%	4%	3%	4%	12%	2%	0	15%	5%
Naive Fusion	**43%**	**18%**	**48%**	**22%**	**55%**	**67%**	**26%**	**4%**	**72%**	**40%**

Table 3, UMT outperforms other methods and does improve multi-modal learning in VGG-Sound and UCF101.

For segmentation. In contrast to the late-fusion classification task, the RGB-Depth semantic segmentation employs middle-fusion architecture. For depth distillation, since features generated by each layer matter, we distill multi-scale depth feature maps using the MSE loss from uni-modal pre-trained depth model to the corresponding parts in multi-modal model. For feature maps from the RGB encoder, however, since they are generated by fusing RGB and depth modalities, we cannot distill RGB feature maps directly like depth feature maps. To mitigate this effect, we curate predictors, namely 2 layers CNNs, aiming to facilitate the fused feature maps to predict the RGB feature maps from uni-RGB trained model. As shown in Table 5, UMT can also improve multi-modal segmentation whether the encoder is pre-trained on ImageNet or not.

Ablation on knowledge distillation. To further verify that the improvement brought by UMT is due to the solving of Modality Laziness, not knowledge distillation, we conduct self-distillation on soft label (Hinton et al., 2015) and feature (Romero et al., 2014), respectively, and compare it with UMT (we also test different objectives for UMT, including L1Loss, MSELoss and Contrastive loss Tian et al. (2019)). As Table 4 shows, naive distillation can only bring limited improvement, which implies Modality Laziness is the most pressing issue.

4.4 Multi-modal training *vs.* uni-modal pre-training

In this subsection, we compare UMT with two uni-modal pre-training approaches, *e.g.*, directly averaging uni-modal models' predictions and training a multi-modal linear classifier on the uni-modal pre-trained encoders. As Table 7 shows, UMT outperforms uni-modal pre-training approaches in VGG-Sound, while directly averaging uni-modal models' predictions in UCF101 gets better accuracy. It is consistent with our findings that VGG-Sound owns more *paired features* than UCF101: as shown in Table 6, in the selected 10 classes of VGG-Sound, naive joint training can achieve better test accuracy than the sum of the two uni-modal models. While in UCF101, there is no one class like those. This also tallies with the motivation of curating VGG-Sound dataset – as much audio-visual synchronization as possible.

Table 7: Comparison of multi-modal training with uni-modal pre-training approaches.

Method	**VGG-Sound**	**UCF101**
Uni-Audio (RGB)	45.15	77.08
Uni-Video (Flow)	23.17	74.99
Avg Prediction	46.10	**86.78**
Linear Classifier	50.95	84.43
UMT	**53.46**	83.72

5 Conclusion

In this paper, we theoretically analyze the modality laziness problem in multi-modal learning, and propose Uni-Modal Teacher (UMT), a distillation-based training approach to remedy this problem. We further conclude that when paired features are rare, combining two individually trained uni-modal models gives good results; otherwise, UMT can achieve competitive results, serving as a pushing force to tackle laziness problem in multi-modal joint training. We hope our findings will shed new light on multi-modal learning research.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Massih R Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views-an application to multilingual text categorization. *Advances in neural information processing systems*, 22:28–36, 2009.

Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Raman Arora, Poorya Mianjy, and Teodor Marinov. Stochastic optimization for multiview representation learning using partial least squares. In *International Conference on Machine Learning*, pp. 1786–1794. PMLR, 2016.

Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29:892–900, 2016.

Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020a.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020c.

Kin Wai Cheuk, Hans Anderson, Kat Agres, and Dorien Herremans. nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks. *IEEE Access*, 8:161981–162003, 2020.

Haytham M Fayek and Anurag Kumar. Large scale audiovisual learning of sounds with weakly labeled data. *arXiv preprint arXiv:2006.01595*, 2020.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1933–1941, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1440–1444. IEEE, 2019.

Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. What makes multimodal learning better than single (provably). *arXiv preprint arXiv:2106.04538*, 2021.

Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (8):1692–1706, 2015.

Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision*, 2018.

Seong-Jin Park, Ki-Sang Hong, and Seungyong Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 4980–4989, 2017.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *arXiv preprint arXiv:2011.09468*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Daniel Seichter, Mona Köhler, Benjamin Lewandowski, Tim Wengefeld, and Horst-Michael Gross. Efficient rgb-d semantic segmentation for indoor scene analysis. *arXiv preprint arXiv:2011.06961*, 2020.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12695–12705, 2020.

Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 570–586, 2018.

A EXPERIMENTAL DETAILS AND ADDITIONAL EXPERIMENTS

A.1 DATASETS

Here, we describe the preprocessing of VGG-Sound, UCF101 and NYU Depth V2 in detail.

VGG-Sound dataset (Chen et al., 2020b), which contains over 200k video clips for 309 different sound classes, is used for evaluating our method. It is an audio-visual dataset *in the wild* where each object that emits sound is also visible in the corresponding video clip, making it suitable for scene classification tasks. Please note that some clips in the dataset are no longer available on YouTube, and we actually use about 175k videos for training and 15k for testing, but the number of classes remains the same. We design a preprocessing paradigm to improve training efficiency as follows: (1) each video is interpolated to 256×256 and saved as stacked images; (2) each audio is first converted to 16 kHz and 32-bit precision in the floating-point PCM format, then randomly cropped or tiled to a fixed duration of 10s. For video input, 32 frames are uniformly sampled from each clip before feeding to the video encoder. While for the audio input, a 1024-point discrete Fourier transform is performed using nnAudio (Cheuk et al., 2020), with 64 ms frame length and 32 ms frame-shift. And we only feed the magnitude spectrogram to the audio encoder.

UCF101 dataset (Soomro et al., 2012) is an action recognition dataset with 101 action categories, including 7k videos for training and 3k for testing. And we use the rgb and flow provided by (Feichtenhofer et al., 2016). For RGB, we use one image of $(3 * 224 * 224)$ as the input; while for flow, we use a stack of optical flow images which contained 10 x-channel and 10 y-channel images, So its input shape is $(20 * 224 * 224)$. During training, we perform random crop and random horizontal flip as the data augmentation; while testing, we resize the image to 224 and do not perform data augmentation operations.

NYU Depth V2 dataset (Silberman et al., 2012) contains 1449 indoor RGB-Depth data totally and we use 40-class label setting. The number of training set and testing set is 795 and 654 respectively. All perprocessing operations are following Seichter et al. (2020).

A.2 TRAINING HYPERPARAMETERS

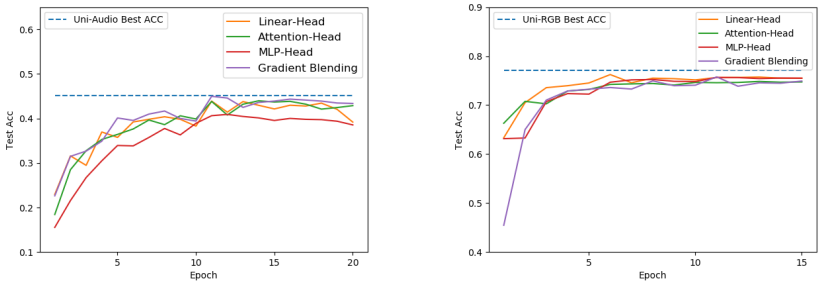
In this subsection, we show the hyperparameters in our experiments in Table 8. For additional losses in Auxiliary-CELoss and Gradient Blending, we use a linear layer to receive the uni-modal features in multi-modal training and generate the losses as the regularizers.

Table 8: The Hyperparameters used in our experiments. Noting that NYU Depth V2's hyperparameters can be found in Seichter et al. (2020) and we use ResNet34 as the backbone.

Hyperparameter	Value (VGG-Sound)	Value (UCF101)
Encoder	ResNet3D (Video), 2D (Audio)	ResNet2D(Both Modalities)
Linear Head	(1024, 309)	(1024, 101)
MLP Head	(1024, 1024)	(1024, 1024)
	ReLU	ReLU
	(1024, 309)	(1024, 101)
Attension Head	Attension Layer (without new parameters) + a linear layer	
Training Epoches	20	20
LR	1e-3	1e-2
Batch Size	24	64
Optimizer	Adam	SGD
Scheduler	StepLR (step=10, gamma=0.1)	ReduceLROnPlateau (patience=1)
Loss Fusion	Cross Entropy for task, MSE for distillation	

A.3 ENCODER EVALUATION IN THE WHOLE TRAINING PROCESS

In this subsection, we show the evaluation of audio encoder on VGG-Sound and RGB encoder on UCF101 in the whole training process. As we can see in Figure 4, all encoders from multi-modal training are bounded by their uni-modal training counterparts.



(a) Audio Encoder Evaluation on VGG-Sound

(b) RGB Encoder Evaluation on UCF101

Figure 4: The evaluation of audio encoder on VGG-Sound and RGB encoder on UCF101 in the whole training process.

A.4 Class-level Modality laziness

As Figure 5 shows, among the classes in which the audio network trained over uni-modal data can achieve good accuracy on VGG-Sound, the video encoder trained by the multi-modal training methods falls behind its uni-modal counterpart. Specifically, the mean accuracy on these classes of three types (Naive Fusion, Gradient Blending, Uni-Video Training) of video encoder are 33.79%, 36.92%, and 49.05% respectively, where the gap between Gradient Blending video and uni-video is 12.13%.

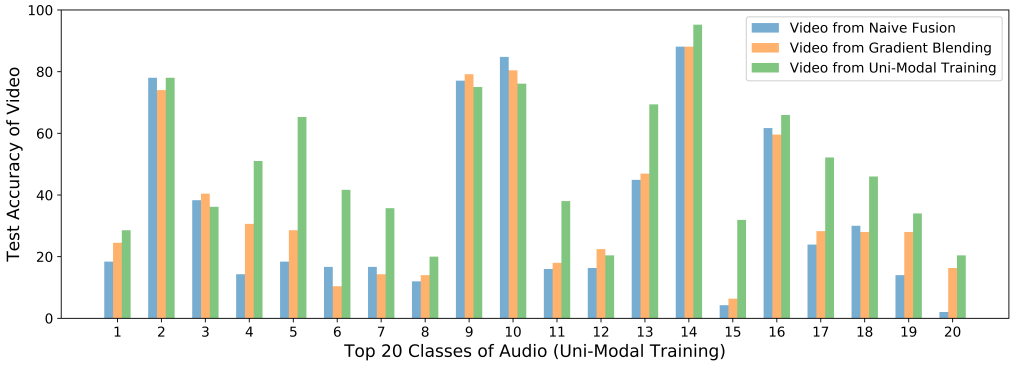


Figure 5: We first select the top 20 test accuracy classes of uni-audio model on VGG-Sound, and then evaluate different video encoders on these classes. It can be seen that the video encoder in multi-modal training setting (linear-head and Gradient Blending) is worse than that in uni-video setting over about 15 classes, indicating that **modality laziness occurs in multi-modal training**.

A.5 UMT in different tasks

In this subsection, we describe how Uni-Molda Teacher (UMT) applies on different multi-modal tasks.

UMT in late-fusion classification. In multi-modal late-fusion architecture, modalities are first encoded by the corresponding encoders and then mapped to the output space (Figure 6 left). Uni-modal Teacher distills the pre-trained uni-modal features to the corresponding parts in multi-modal networks in multi-modal training (Figure 6 right).

UMT in multi-modal middle-fusion segmentation. In contrast to the late fusion classification task, the RGB-Depth semantic segmentation belongs to middle fusion. The main encoder receives RGB inputs, and the depth inputs are fed into the depth encoder. At each intermediate layer, the

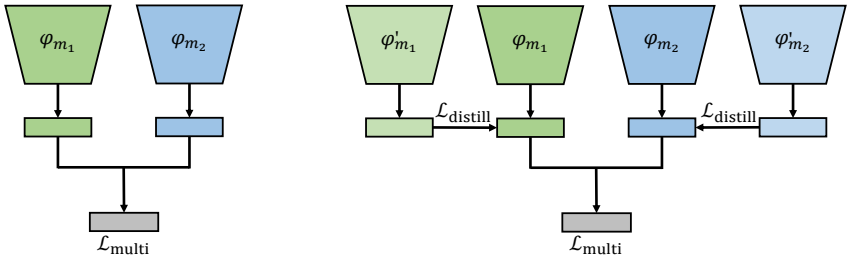


Figure 6: Model architecture of naive late fusion (left) and Uni-Modal Teacher (UMT) (right).

main encoder fuses its own intermediate outputs and the depth features obtained from the depth encoder, which makes it a mid-fusion task (Seichter et al., 2020). Since features generated by each layer matter, we distill multi-scale depth feature maps using the MSE loss. For feature maps from the RGB encoder, however, since they are generated by fusing RGB and depth modalities, we cannot distill RGB feature maps directly like depth feature maps. To mitigate this effect, we curate predictors, namely 2 layers CNNs, aiming to facilitate the fused feature maps to predict the RGB feature maps trained by the RGB modality before distillation. The full schematic diagram is presented in Figure 7.

UMT's weights. For VGG-Sound, we use 50 (both audio feature distillation and video feature distillation) as the distillation loss's weight; for NYU Depth V2, we use 1 as the distillation weight (both RGB and Depth), and multiply it by 0.1 every 100 epochs. Because the amount of data in UCF101 is small, it is more sensitive to the weights, and we set 20 as the rgb distillation weight and 0.1 as the flow distillation weight.

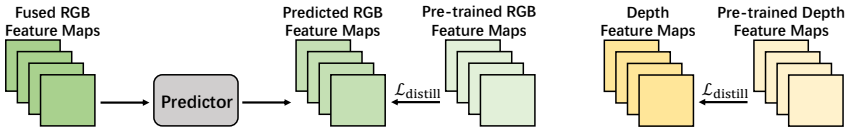


Figure 7: Distillation details of UMT for RGB (left) and depth (right) modalities in multi-modal semantic segmentation (based on ESANet).

A.6 Finetuning the uni-modal pre-trained encoders in multi-modal training

Uni-modal pre-training approaches aim to combine two trained uni-modal encoders without updating the parameters of them. Here, we use the uni-modal pre-trained encoders' parameters as the initialized weights in multi-modal training and jointly fine-tune the encoders and a new multi-modal linear classifier on VGG-Sound. We set the classifier's learning rate as $1e-3$. As Figure 8 and Table **??** show, using the uni-modal pre-trained weights in multi-modal training and then fine-tuning the encoders cannot bring significant improvement. When the learning rate is large, the encoders forget some abilities to extract self-standing features (see Table **??**).

A.7 Can Mid-Fusion Tackle Modality Laziness?

In this subsection, we test a different multi-modal fusion approach, middle fusion. Specifically, we use the average pool on the third block's outputs of the video encoder (VGG-Sound), and then tile them to get the same shape as audio feature maps. Noting that the audio feature maps are also the output of third block in the audio encoder. Then we concatenate two groups of feature maps before the last block and the output layer, which is similar to Owens & Efros (2018). As Table 10 shows, middle fusion is significantly worse than UMT, which implies that middle fusion approach also suffer from *Modality Laziness*.

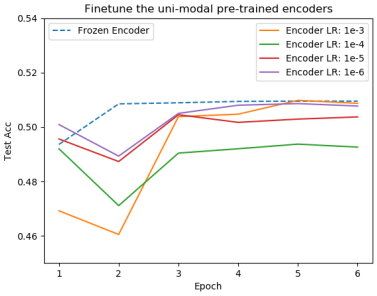


Figure 8: Finetuning Process

Encoder LR	Top-1 Acc	Encoder Eval	
		Audio	Video
1e-3	50.98	43.98	21.86
1e-4	49.37	44.71	21.97
1e-5	50.45	45.28	23.13
1e-6	50.86	45.29	23.27
0	50.95	45.15	23.17

Table 9: Finetuning Results on VGG-Sound

Table 10: Middle Fusion vs Late Fusion (VGG-Sound)

Method	Late Fusion	Middle Fusion	UMT (based on Late Fusion)
Top-1 Accuracy	49.46	49.87	53.46

A.8 Can Dropout Tackle Modality Laziness?

Here we consider the common regularizer, dropout (Srivastava et al., 2014), and a variant of it, namely modality-wise dropout, which randomly drops (with probability $1/3$) the feature from one modality in every iteration. Modality dropout is akin to the ModDrop in Neverova et al. (2015). As Table 11 shows, modality-wise dropout is significantly better than dropout, which implies that modality-wise laziness is serious and modality-wise dropout is effective.

Table 11: Dropout in multi-modal training (VGG-Sound)

Method	Naive Fusion	Dropout	Modality Dropout	UMT
Top-1 Accuracy	49.46	49.83	51.37	53.46

A.9 Sensitivity Analysis on Distillation Weights

Here we test different distillation weights on VGG-Sound. As shown in Table 12, if the weight is too small, the model will lack of strength to fight against Modality Laziness; on the other hand, if the weight is too large, the weight of multi-modal cross entropy loss would be relatively small, which hinders joint multi-modal feature learning.

A.10 Paired Feature on VGG-Sound and UCF101

As we can see in Table 6, the multi-modal naive fusion model outperforms other uni-modal models in some selected classes, and even exceeding the sum of the accuracy of the uni-audio model and uni-video model, which implies the importance of paired features in VGG-Sound. Although there are some classes that naive fusion outperforms uni-modal models on UCF101, we cannot find any classes that naive fusion can exceed the sum of the accuracy of the uni-RGB model and uni-flow model and the improvement brought by naive fusion is also limited, as shown in Table 13.

The results show that **VGG-Sound has a larger proportion of paired features than UCF101.**

The correspondence between id and name of the selected class in VGG-Sound is: 164: People Sniggering, 303: Wood Thrush Calling, 33: Cat Meowing, 255: Sea Waves, 91: Footsteps On Snow, 4: Alligators Crocodiles Hissing, 152: People Gargling, 127: Mynah Bird Singing, 68: Door Slamming, 155: People Humming.

Table 12: Different distillation weights of UMT on VGG-Sound

Weights	0	1	10	20	50	100
Top-1 Accuracy	49.46	49.51	51.31	51.51	53.46	53.11

Table 13: Top-1 test accuracy of different models on selected classes of UCF101. We select the top-10 classes according to the gap of accuracy between the multi-modal and uni-modal models.

Class ID	54	0	13	12	48	50	6	80	22	34
Uni-RGB	70%	95%	91%	76%	78%	72%	74%	73%	61%	45%
Uni-Flow	38%	68%	69%	58%	64%	54%	60%	53%	47%	24%
Naive Fusion	73%	100%	100%	87%	92%	79%	86%	78%	72%	48%

B PROOF

B.1 PROOF OF THEOREM 1

Theorem 1 *Assume that in uni-modal pre-training approaches, the number of features learned in modality x^{m_1} is b_1 and the number of features learned in modality x^{m_2} is b_2. We order the probability of self-standing features (both x^{m_1} and x^{m_2}) in decreasing order of p, namely, $p_{[1]}, \ldots, p_{[i]}$. Assume that multi-modal training approaches learn k_1 self-standing features in modality x^{m_1}, k_2 self-standing features in modality x^{m_2}, and k_3 paired features with predicting probability $p(h_1), \ldots, p(h_{k_3})$. Then the following statements hold:*

- *(a.) **Quantity Laziness**: $k_1 + k_2 + k_3 \leq \min\{b_1, b_2\}$.*
- *(b.) **Uni-modal Laziness**: Each modality in multi-modal training approaches performs worse than uni-modal approaches.*
- *(c.) **Performance Laziness**: Consider a new testing point, for every $\delta > 0$, if $\sum_{i\in[k_3]} p(h_i) \leq \sum_{i\in[b_1+1,b_1+b_2]} p_{[i]} + \sqrt{8(k_3 + b_1 - k_1 + b_2 - k_2)\log(1/\delta)}$, then uni-modal pre-training approaches outperform multi-modal training approaches concerning the loss on the testing point with probability at least $1 - \delta$, where the probability is taken over the randomness of the testing point.*

We prove the next theorem, which shows that multi-modal training approaches indeed suffers from overfitting issues, meaning that it learns less features compared to uni-modal pre-training approaches.

Proof: We first introduce some additional notations used in the proof. We define the features trained in x^{m_1}-uni-modal training as $f_1(x^{m_1}), \ldots, f_{b_1}(x^{m_1})$, define the features trained in x^{m_1}-uni-modal training as $g_1(x^{m_2}), \ldots, g_{b_2}(x^{m_2})$. Therefore, there are in total $b_1 + b_2$ features learned in uni-modal pre-training approaches, namely, $f_1(x^{m_1}), \ldots, f_{b_1}(x^{m_1}), g_1(x^{m_2}), \ldots, g_{b_2}(x^{m_2})$. Besides, We define the features trained in multi-modal training approaches as $f_1(x^{m_1}), \ldots, f_{k_1}(x^{m_1})$, $g_1(x^{m_2}), \ldots, g_{k_2}(x^{m_2})$, $h_1(x^{m_1}, x^{m_2}), \ldots, h_{k_3}(x^{m_1}, x^{m_2})$. When the context is clear, we omit the dependency of x^{m_1}, x^{m_2} and denote them as f_i, g_i, h_i for simplicity. When the context is clear, we abuse the notation r to represent arbitrary f, g or h. The corresponding predicting probability of feature r_i is denoted as $p(r_i)$. To summary, there are $b_1 + b_2$ features in uni-modal pre-training approaches, $k_1 + k_2 + k_3$ features in multi-modal training approaches.

We first prove statement (a.), which claims that the number of features learned in multi-modal training approaches are provably less than any of the number of features learned in uni-modal training. The proof depends on the following Lemma 1.

Lemma 1 *Assume there exists T features $r_i, i = 1, \ldots, T$. If we replace one of the T features (without loss of generality, r_T) with a more powerful feature r', where $p(r') > p(r_T)$, then the predicting probability for each data point increases (where the probability is taken over the randomness of the training data).*

We next provide the proof of statements (a.): based on Lemma 1. We shall prove $k_1 + k_2 + k_3 < b_1$ without loss of generality. Start from the features $f_1(x^{m_1}), \ldots, f_{k_1}(x^{m_1})$ which are common features in both multi-modal training approaches and Uni-modal training. Next step, we add feature f_{k_1+1} in uni-modal approachesand g_1 in multi-modal training approaches. Obviously, $p(g_1) > p(f_{k_1+1})$ due to the training priority (or multi-modal training approaches should learn f_{k_1+1} instead of g_1). Therefore, the predicting probability of multi-modal training approaches is larger than uni-modal approaches.

Repeating the procedure by comparing g_i with f_{k_1+i} and comparing h_j with $f_{k_1+k_2+j}$, the predicting probability of multi-modal training approaches is always larger than uni-modal approaches. Note that b_1 should be always larger than $k_1 + k_2$, or the predicting probability of uni-modal approaches would be smaller than multi-modal training approaches. At the end of the comparison, the predicting probability of multi-modal training approaches is still larger than uni-modal approaches. This requires that uni-modal approaches should learn more features, which can be regarded as uni-modal approacheslearns a features while multi-modal training approaches learns an empty feature. In conclusion, uni-modal approaches learns more features compared to multi-modal training approaches, leading to $b_1 > k_1 + k_2 + k_3$.

We next prove the statement (b.). The proof of (b.) is based on (a.). We next only consider modality x^{m_1}, the proof for modality x^{m_2} is similar. Note that the since the number of features learned in multi-modal training approaches is less than b_1, the number of features learned in x^{m_1} must be less than b_1 (Note that those features can be either paired feature or self-standing feature, namely, $f_1, \ldots, f_{k_1}$ and $h_1, \ldots, h_{k_3}$). Therefore, multi-modal training approaches learns less features compared to uni-modal approaches in modality x^{m_1}. On the other hand, the predicting probability of features learned in multi-modal training approaches ($f_1, \ldots, f_{k_1}$ and $h_1, \ldots, h_{k_3}$, considering only modality x^{m_1} for the paired feature) is less than that learned in uni-modal approaches ($f_1, \ldots, f_{b_1}$), because otherwise, uni-modal approaches will learn the features in h instead of f. In conclusion, when considering only modality x^{m_1}, the number of features learned in multi-modal training approaches is less and its corresponding predicting probability is small. Therefore, each modality in multi-modal training approaches performs worse than uni-modal approaches.

We finally prove the statement (c.). Recall that the loss is $-\sum_i u(r_i)$ where $u(r_i) = \mathbb{I}(yr_i > 0) - \mathbb{I}(yr_i < 0)$. Note that $\mathbb{E}(u(r_i)) = \frac{1}{2}p(r_i)$ and $|u(r_i)| \leq 1$. We derive that:

$$
\begin{aligned}
&\mathbb{P}\left(-\sum_{i\in[k_1]} u(f_i) - \sum_{i\in[k_2]} u(g_i) - \sum_{i\in[k_3]} u(h_i) \leq -\sum_{i\in[b_1]} u(f_i) - \sum_{i\in[b_2]} u(g_i)\right) \\
=&\mathbb{P}\left(\sum_{k_1<i\leq b_1} u(f_i) + \sum_{k_2<i\leq b_2} u(g_i) - \sum_{i\in[k_3]} u(h_i) \leq 0\right) \\
=&\mathbb{P}\left(\sum_{k_1<i\leq b_1} u(f_i) + \sum_{k_2<i\leq b_2} u(g_i) - \sum_{i\in[k_3]} u(h_i) + \frac{1}{2}E \leq \frac{1}{2}E\right),
\end{aligned}
$$

where $E = -\mathbb{E}(\sum_{k_1<i\leq b_1} u(f_i) + \sum_{k_2<i\leq b_2} u(g_i) - \sum_{i\in[k_3]} u(h_i)) = \sum_{i\in[k_3]} p(h_i) - \sum_{k_1<i\leq b_1} p(f_i) - \sum_{k_2<i\leq b_2} p(g_i)$. Due to the training priority and the conclusion in (a.),

$$
\sum_{i\in[b_1+1,b_1+b_2]} p_{[i]} \leq \sum_{k_1<i\leq b_1} p(f_i) + \sum_{k_2<i\leq b_2} p(g_i).
$$

Therefore, $E \leq \sum_{i\in[k_3]} p(h_i) - \sum_{i\in[b_1+1,b_1+b_2]} p_{[i]} \leq \sqrt{8(k_3 + b_1 - k_1 + b_2 - k_2)\log(1/\delta)}$. We next apply Hoeffding inequality on Equation B.1 and derive that

$$
\begin{aligned}
&\mathbb{P}\left(-\sum_{i\in[k_1]} u(f_i) - \sum_{i\in[k_2]} u(g_i) - \sum_{i\in[k_3]} u(h_i) < -\sum_{i\in[b_1]} u(f_i) - \sum_{i\in[b_2]} u(g_i)\right) \\
\leq& \exp(-E^2/8(k_3 + b_1 - k_1 + b_2 - k_2)) \\
\leq& \delta
\end{aligned}
$$

To conclude, multi-modal training approaches outperform uni-modal pre-training approaches concerning the testing loss with probability at least $1-\delta$.

Compared to uni-modal pre-training approaches, denote the additional paired feature are indexed by c, and the additional self-standing feature in uni-modal pre-training approaches are indexed by v. We have that:

$$\begin{aligned}
&\mathbb{P}(\sum_{i\in[c]}(I(f_i(x)>0)-I(f_i(x)<0))-\sum_{j\in[v]}(I(f_j(x)>0)-I(f_j(x)<0))>0)\\
=&\mathbb{P}(\sum_{i\in[c]}I(f_i(x)>0)-\sum_{j\in[v]}I(f_j(x)>0)-\frac{1}{2}[\sum_{i\in[c]}p_i-\sum_{j\in[v]}p_j]>\frac{1}{2}[\sum_{j\in[v]}p_j-\sum_{i\in[c]}p_i]) \quad (1)\\
\leq&\exp(-(\sum_{j\in[v]}p_j-\sum_{i\in[c]}p_i)^2/8|c+v|)
\end{aligned}$$

Therefore, if $\sum_{j\in[v]}p_j-\sum_{i\in[c]}p_i\geq\sqrt{8(c+v)\log(1/\delta)}$, the probability is done. Therefore, for a new data point, uni-modal pre-training approaches can outperforms multi-modal training approaches with high probability.

□

Proof:[Proof of Lemma 1] We define $r_{[-T]}$ as the features $r_1,\ldots,r_{T-1}$. The proof is divided into two parts, depending on whether $\sum_{i\in[T-1]}\mathbb{I}(r_i\neq 0)$ is even or odd. We regard the term $\sum_{i\in[T-1]}\mathbb{I}(r_i\neq 0)$ as the number of effective features in $r_{[-T]}$. To simplify the discussion, we rescale r such that $|yr|=1$ (when $r\neq 0$) or $|yr|=0$ (when $r=0$).

Case 1: When the number of effective features in $r_{[-T]}$ is even. (a.) If $|\sum_{i\in[T-1]}yr_i|\geq 2$, adding r_T or r' does not alter the predicting probability, namely

$$\begin{aligned}
&\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]}yr_i\right]>0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|\geq 2\right)+\frac{1}{2}\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]}yr_i\right]=0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|\geq 2\right)\\
=&\mathbb{P}\left(y\left[r'+\sum_{i\in[T-1]}yr_i\right]>0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|\geq 2\right)+\frac{1}{2}\mathbb{P}\left(y\left[r'+\sum_{i\in[T-1]}yr_i\right]=0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|\geq 2\right).
\end{aligned}$$

(b.) When the number of effective features in $r_{[-T]}$ is even, $|\sum_{i\in[T-1]}yr_i|\neq 1$.

(c.) When $|\sum_{i\in[T-1]}yr_i|=0$, due to the assumption that $p(r')>p(r_T)$ and $\epsilon(r)=p(r)/c$, adding r' helps increase the predicting probability compared to r_T, namely

$$\begin{aligned}
&\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]}yr_i\right]>0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)+\frac{1}{2}\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]}yr_i\right]=0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)\\
>&\mathbb{P}\left(y\left[r'+\sum_{i\in[T-1]}yr_i\right]>0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)+\frac{1}{2}\mathbb{P}\left(y\left[r'+\sum_{i\in[T-1]}yr_i\right]=0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right).
\end{aligned}$$

The above inequality is derived based on the following equation:

$$\begin{aligned}
&\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]}yr_i\right]>0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)+\frac{1}{2}\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]}yr_i\right]=0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)\\
=&\mathbb{P}\left(yr_T>0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)+\frac{1}{2}\mathbb{P}\left(yr_T=0\,\middle|\,\left|\sum_{i\in[T-1]}yr_i\right|=0\right)\\
=&p(r_T)+\frac{1}{2}\left[1-p(r_T)-\epsilon(r_T)\right]\\
=&\frac{1}{2}\left[1+(1-1/c)p(r_T)\right].
\end{aligned}$$

Since we assume $c>1$, the probability increases with probability $p(r_T)$.

Therefore, under the three conditions, adding r' increase the predicting probability more compared to r_T. In summary, under case 1 (a-c), adding r' increase the predicting probability compared to r_T.

Case 2: When the number of features in $r_{[-T]}$ is odd. The discussion in (b.) can be a little bit more complex compared to case 1.

(a.) If $|\sum_{i\in[T-1]} yr_i| \geq 2$, similar to case 1, adding r_T or r' does not alter the predicting probability, namely

$$\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]} yr_i\right]>0 \,\middle|\, \left|\sum_{i\in[T-1]} yr_i\right|\geq 2\right)+\frac{1}{2}\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]} yr_i\right]=0 \,\middle|\, \left|\sum_{i\in[T-1]} yr_i\right|\geq 2\right)$$
$$=\mathbb{P}\left(y\left[r'+\sum_{i\in[T-1]} yr_i\right]>0 \,\middle|\, \left|\sum_{i\in[T-1]} yr_i\right|\geq 2\right)+\frac{1}{2}\mathbb{P}\left(y\left[r'+\sum_{i\in[T-1]} yr_i\right]=0 \,\middle|\, \left|\sum_{i\in[T-1]} yr_i\right|\geq 2\right).$$

(b.) If $|\sum_{i\in[T-1]} yr_i| = 1$: (b.1) If $\sum_{i\in[T-1]} yr_i = -1$:

$$\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]} yr_i\right]>0 \,\middle|\, \sum_{i\in[T-1]} yr_i=-1\right)+\frac{1}{2}\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]} yr_i\right]=0 \,\middle|\, \sum_{i\in[T-1]} yr_i=-1\right)$$
$$=\mathbb{P}\left(yr_T-1>0 \,\middle|\, \sum_{i\in[T-1]} yr_i=-1\right)+\frac{1}{2}\mathbb{P}\left(yr_T-1=0 \,\middle|\, \sum_{i\in[T-1]} yr_i=-1\right)$$
$$=\frac{1}{2}\mathbb{P}\left(yr_T-1=0 \,\middle|\, \sum_{i\in[T-1]} yr_i=-1\right)$$
$$=\frac{1}{2}p(r_T).$$

(b.2) If $\sum_{i\in[T-1]} yr_i = +1$:

$$\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]} yr_i\right]>0 \,\middle|\, \sum_{i\in[T-1]} yr_i=1\right)+\frac{1}{2}\mathbb{P}\left(y\left[r_T+\sum_{i\in[T-1]} yr_i\right]=0 \,\middle|\, \sum_{i\in[T-1]} yr_i=1\right)$$
$$=\mathbb{P}\left(yr_T+1>0 \,\middle|\, \sum_{i\in[T-1]} yr_i=1\right)+\frac{1}{2}\mathbb{P}\left(yr_T+1=0 \,\middle|\, \sum_{i\in[T-1]} yr_i=1\right)$$
$$=(1-\epsilon(r_T))+\frac{1}{2}\epsilon(r_T)$$
$$=1-\frac{1}{2c}p(r_T).$$

Note that the probability of event (b.1) and the probability of event (b.2) satisfy the following equation by Lemma 2:

$$\mathbb{P}\left(\sum_{i\in[T-1]} yr_i=1\right)=c\mathbb{P}\left(\sum_{i\in[T-1]} yr_i=-1\right). \tag{2}$$

Therefore, the total probability under case (b) is

$$\frac{1}{2}p(r_T)\mathbb{P}\left(\sum_{i\in[T-1]} yr_i=-1\right)+(1-\frac{1}{2c}p(r_T))\mathbb{P}\left(\sum_{i\in[T-1]} yr_i=1\right)$$
$$=\mathbb{P}\left(\sum_{i\in[T-1]} yr_i=1\right)$$

which is independent of $p(r_T)$. Therefore, adding r_T or r' share the same predicting probability.

(c.) When the number of effective features in $r_{[-T]}$ is odd, $|\sum_{i\in[T-1]} yr_i| \neq 0$.

In summary, under case 2 (a-c), adding r' do not decrease the predicting probability compared to r_T.

The following lemmas are used during the proof.

Lemma 2 *Consider $T-1$ features $r_1, \ldots, r_{T-1}$, the following equation holds:*

$$\mathbb{P}\left(\sum_{i\in[T-1]} yr_i = 1\right) = c\mathbb{P}\left(\sum_{i\in[T-1]} yr_i = -1\right). \tag{3}$$

Proof: It can be proved to compare the events $A = \{\sum_{i\in[T-1]} yr_i = 1\}$ and $B = \{\sum_{i\in[T-1]} yr_i = -1\}$. Every event in A has a complementary event in B, namely,

$$\begin{aligned} yr_i = 1 \text{ in } B \text{ if } yr_i = -1 \text{ in } A \\ yr_i = -1 \text{ in } B \text{ if } yr_i = 1 \text{ in } A \\ yr_i = 0 \text{ in } B \text{ if } yr_i = 0 \text{ in } A \end{aligned}$$

Comparing each event in A with its complementary event in B leads to the conclusion. □

Combining case 1 and case 2 together leads to the final conclusion.

□

B.2 Proof of Theorem 2

Theorem 2 *Denote the paired features by $h_1, \ldots h_L$ with corresponding predicting probability $p(h_1), \ldots, p(h_L)$. Assume that distillation can boost the training priority by $p^0 > 0$. If there exists paired feature whose predicting probability exceeds the boosting probability p^0, namely, the set $\mathcal{S}$ is not empty:*

$$\mathcal{S} = \{h_i : p(h_i) > p^0\} \neq \phi.$$

Then UMT can learn paired feature which cannot be learned by uni-modal pre-training approaches, namely, UMT outperforms uni-modal pre-training approaches.

Proof: The core of Theorem 2 is to clarify the training priority. We revisit the notations of Theorem 1 as follows without further clarification. At the end of the training, uni-modal pre-training approaches learn $b_1 + b_2$ useful features, namely, $f_1, \ldots, f_{b_1}, g_1, \ldots, g_{b_2}$. And multi-modal training approaches learn $k_1 + k_2 + k_3$ features: $f_1, \ldots, f_{k_1}, g_1, \ldots, g_{k_2}, h_1, \ldots, h_{k_3}$. We note that there are still many empty features e_i in the model due to the initialization.

By distillation, the model learns the features according to the new priority. Since the set $\mathcal{S}$ is not empty, there exists paired features that is learned before the empty features. By distillation, the model would learn all the useful features that appear in uni-modal approaches, as well as those features in set $\mathcal{S}$. Therefore, UMT outperforms uni-modal pre-training approaches.

We additionally remark that UMT may lose some paired features compared to multi-modal training approaches. However, multi-modal training approaches learn less self-standing features compared to UMT due to modality laziness. □

B.3 A concrete example to illustrate Theorem 1

We next provide a concrete example to better illustrate the Modality Laziness issues. For Example 1, we aim to show the Modality Laziness issues. For Example 2, we aim to show the role of the pushing force.

Example 1 *Consider modality x^{m_1} with features f_1, f_2, f_3 (corresponding prediction probability $p = 0.2, 0.1, 0.05$), and modality x^{m_2} with features g_1, g_2, g_3 (corresponding prediction probability $p = 0.15, 0.08, 0.02$). We show the dataset in Table 14 and aim to minimize the training loss to zero.*

	f_1	f_2	f_3	g_1	g_2	g_3	h	y
p	0.20	0.10	0.05	0.15	0.08	0.02	0.28	/
p'	0.35(↑)	0.25(↑)	0.20(↑)	0.32(↑)	0.23(↑)	0.17(↑)	0.28	/
data a	+	+	+	+	-	+	+	+1
data b	0	+	0	+	+	-	+	+1
data c	+	+	0	-	+	+	0	-1
data d	+	-	+	+	+	0	+	-1

Table 14: Dataset used in Example 1. $+$ means the feature is larger than zero and $-$ means the feature is less than zero. We denote the predicting probability by p and the rectified probability (due to pushing force) by p'.

In uni-modal approaches, we learn features f_1, f_2 and f_3 on modality x^{m_1} (similarly, g_1, g_2, and g_3 on modality x^{m_2}). Therefore, we learn features $f_1, f_2, f_3, g_1, g_2, g_3$ in uni-modal pre-training approaches. In multi-modal training approaches without paired feature, we can only learn three features f_1, f_2, g_2 due to the training priority $f_1 > g_1 > f_2 > g_2 > f_3 > g_3$ (decreasing order in p). This phenomenon is caused by modality laziness.

We next consider another paired feature h with probability $p = 0.28$. Under the case, multi-modal training approaches only learn two features h and f_1. Therefore, when h is not powerful enough, uni-modal pre-training approaches outperforms multi-modal training approaches.

Example 2 *We follow the notations and dataset in Example 1. By applying the pushing force, assume that each probability of self-standing feature boosts 0.15, which changes the training priority to $f_1 > g_1 > h > f_2 > g_2 > f_3 > g_3$ (decreasing order in p'). Therefore, multi-modal training approaches (with pushing force) learns f_1, f_2, h. As a comparison, multi-modal training approaches (without pushing force) can only learn f_1, h. Therefore, pushing force helps learn more features. We additionally remark that we only consider the training error in this example, and there might be other penalties in practice (e.g., distillation loss).*