

Dynamic Multimodal Fusion

Zihui Xue Radu Marculescu
The University of Texas at Austin

Abstract

Deep multimodal learning has achieved great progress in recent years. However, current fusion approaches are static in nature, i.e., they process and fuse multimodal inputs with identical computation, without accounting for diverse computational demands of different multimodal data. In this work, we propose dynamic multimodal fusion (DynMM), a new approach that adaptively fuses multimodal data and generates data-dependent forward paths during inference. To this end, we propose a gating function to provide modality-level or fusion-level decisions on-the-fly based on multimodal features and a resource-aware loss function that encourages computational efficiency. Results on various multimodal tasks demonstrate the efficiency and wide applicability of our approach. For instance, DynMM can reduce the computation costs by 46.5% with only a negligible accuracy loss (CMU-MOSEI sentiment analysis) and improve segmentation performance with over 21% savings in computation (NYU Depth V2 semantic segmentation) when compared with static fusion approaches. We believe our approach opens a new direction towards dynamic multimodal network design, with applications to a wide range of multimodal tasks.¹

1. Introduction

Humans perceive the world in a multimodal way, through vision, hearing, touch, taste, etc. Recent years have witnessed great progress of deep learning approaches that leverage data of multiple modalities. Consequently, multimodal fusion has boosted the performance of many classical problems, such as sentiment analysis [21, 38, 50], action recognition [6, 36], or semantic segmentation [35, 45].

Despite these advances, how to best combine information characterized by multiple modalities remains a fundamental challenge in multimodal learning [2]. Various research efforts [14, 20, 25, 26, 29, 42, 43, 50] have been put into designing new fusion paradigms that can effectively fuse multimodal data. These approaches are generally task-

¹Our code is available at <https://github.com/zihuiXue/DynMM>.

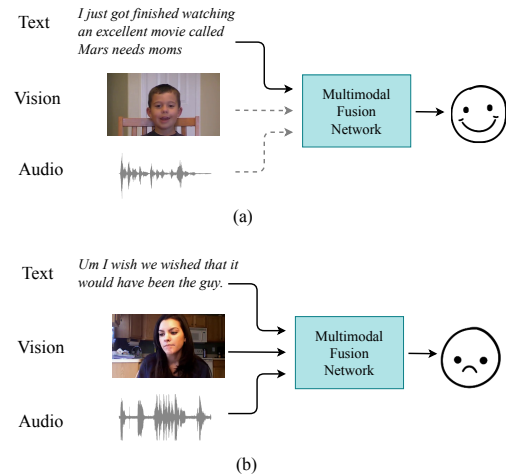


Figure 1. Two examples in CMU-MOSEI [51] for emotion recognition. Figure (a) shows an “easy” multimodal instance as using textual information is sufficient to predict emotions correctly (this is a positive emotion). Figure (b) shows a “hard” example where all three modalities are required to make correct predictions (this is a negative emotion). While static multimodal fusion networks process “hard” and “easy” inputs identically, we propose *dynamic instance-wise inference* that can achieve computational savings for “easy” examples and preserve representation power for “hard” instances. For (a), DynMM only activates the text path and skips paths corresponding to the other two modalities, thus leading to computational efficiency.

and modality-specific and require manual design. Building on the success of Neural Architecture Search (NAS), a few recent works [33, 39, 49] have adopted NAS to find effective fusion architectures automatically.

However, both manually-designed and NAS-based approaches process all the instances in a *single* fusion architecture and lack adaptability to diverse multimodal data. Namely, once the fusion network is trained, it performs static inference on each piece of data, without accounting for the inherent differences in characteristics of different multimodal inputs. Thus, the computational efficiency, as well as the representation power of a well-designed fusion architecture may be limited by its static nature. As a motivating example, consider the two multimodal instances in

Figure 1. As shown, it is relatively easy to classify emotions for the upper example: the text modality alone provides strong evidence for a positive emotion. On the other hand, it is unlikely to correctly predict emotions for the lower example based solely on the textual information since this sentence is confusing. Audio and visual modalities can provide important cues to a multimodal network to make correct decisions. From this example, we can see that multimodal data enable a model to learn from the rich representations of “hard” inputs; it can also bring redundancy in computations for the “easy” inputs.

Inspired by this observation, we propose *dynamic multimodal fusion (DynMM)*, a new approach that *adaptively fuses* input data from multiple modalities. Compared with a static multimodal architecture, DynMM enjoys the benefits of reduced computation, improved representation power and robustness. More precisely, dynamic fusion leads to computational savings for “easy” inputs that can be correctly predicted using only a subset of modalities or simple fusion operations. For “hard” multimodal inputs, DynMM can match the representation power of a static network by relying on all modalities and complex fusion operations for prediction. In addition, real-world multimodal data may be noisy and contradictory [22]. In such cases, skipping paths that involve noisy modalities for certain instances in DynMM can reduce noise and boost performance.

Dynamic neural networks [11] have gained increasing attention over the past few years and enjoys a broad range of applications, such as image recognition [5, 28, 44, 46], semantic segmentation [23, 41] and machine translation [37]. Motivated by the great success of dynamic inference for unimodal networks, this paper aims at proposing multimodal fusion as a new application domain. To this end, we draw inspiration from the natural redundancy of multimodal data, which provides a different angle from existing work. To be specific, we propose *progressive fusion*, both at *modality level* and at *fusion level*. At *modality level*, we train a gating network to select a subset of input modalities (or all modalities) for predictions based on each input. At *fusion level*, the gating network provides sample-wise decisions on which fusion operation to adopt and when to stop fusion. On one hand, by allowing exits at the early fusion stages for “easy” inputs, DynMM saves the computations of executing the later fusion modules. On the other hand, in terms of “hard” multimodal inputs, DynMM can turn all fusion modules on for accurate predictions.

To verify the efficacy and generalizability of our approach, we conduct experiments on various popular multimodal tasks. DynMM strikes a good balance between computational efficiency and learning performance. For instance, for RGB-D semantic segmentation tasks, DynMM achieves a +0.7% mIoU improvement with over 21% reductions in multiply-add operations (MAdds) for the depth

encoder when compared against [35]. Moreover, we find that DynMM yields better predictions than static fusion networks when the input modality is perturbed by noise; this suggests possible use of DynMM to improve the multimodal robustness.

2. Related Work

2.1. Dynamic Neural Networks

Dynamic neural networks have demonstrated a great potential in classical computer vision problems, such as image classification [5, 28, 44, 46], object detection [7, 52], or semantic segmentation [23, 41]. While popular deep learning approaches perform inference in a static manner, dynamic networks allow the network structure to adapt to the input characteristics during inference. This flexibility yields many benefits, including high efficiency, representation power and results interpretability [10, 34, 47]. Dynamic network designs can be categorized into: (a) dynamic depth; (b) dynamic width; (c) dynamic routing [11].

The idea of dynamic depth is to adjust the network depth based on each sample. By providing early exits [4, 40] in shallow layers, one can save computations by not activating deep layers for “easy” samples. For dynamic width, the idea is to adapt the network width in a sample-wise manner. To build a dynamic width network and achieve inference efficiency, previous works have proposed to skip neurons in fully-connected layers [3], skip branches in Mixture-of-Experts (MoE) [28, 37], or skip channels in Convolutional Neural Networks (CNNs) [17]. To enable more flexibility, recent works [5, 23] build SuperNets with multiple inference paths. Dynamic routing is thus performed inside the SuperNet to generate data-dependent forward paths during inference. Our proposed modality-level DynMM belongs to the category of *dynamic width* approaches; the fusion-level DynMM can be seen as a *dynamic routing* approach.

2.2. Multimodal Learning

Multimodal fusion networks have a clear advantage over their unimodal counterparts in various applications, such as sentiment analysis [21, 38, 50], action recognition [6, 36], or semantic segmentation [8, 35, 45]. However, how to effectively combine multimodal features to better exploit information remains a big challenge. Existing works either propose hand-crafted fusion designs based on domain knowledge [20, 25, 26, 29, 43, 50], or apply NAS to find good architectures automatically [33, 39, 49]. However, the scope of these works is limited to static networks only.

There have been some early attempts in adopting dynamic neural networks for multimodal applications, such as semantic segmentation [45], video recognition [9, 32], visual-inertial odometry [48] and medical classification [12]. Among them, CEN [45] dynamically exchanges

channels between sub-networks of the RGB and depth modality for performance improvement. Han *et al.* [12] proposes to dynamically evaluate feature-level and modality-level informativeness of different samples for more trustworthy medical classification, yet the angle of computational efficiency brought by the dynamic neural networks is overlooked. The work of Gao *et al.* [9] and AdaMML [32] are most relevant to our approach as they also adaptively utilize modalities for efficient video recognition. However, their methods are tailored for video data and action recognition. In this work, we aim to make the first step towards a systematic and general formulation of dynamic multimodal fusion that can suit various multimodal tasks.

3. Method

In this section, we present the key design contributions of our proposed dynamic multimodal fusion network (DynMM). First, we introduce new decision making schemes that enable DynMM to generate data-dependent forward paths during inference. Two levels of granularity are considered, *i.e.*, modality-level (coarse level) and fusion-level (fine level) decision making. Next, we propose new training strategies for DynMM, which consist of (1) a training objective that accounts for resource budgets, and (2) optimization of a non-differentiable gating network.

3.1. Modality-level Decision

Assume that input data has M modalities, denoted by $\mathbf{x} = (x_1, x_2, \dots, x_M)$. Following the classical Mixture-of-Experts (MoE) [27] framework, we design a set of expert networks as follows. Each expert specializes in a subset of all M modalities. If $M = 3$, for example, we can have up to 7 expert networks, denoted by $E_1(x_1)$, $E_2(x_2)$, $E_3(x_3)$, $E_4(x_1, x_2)$, $E_5(x_2, x_3)$, $E_6(x_1, x_2)$, $E_7(x_1, x_2, x_3)$. In real applications, the candidate expert networks can be narrowed down with domain expertise. For instance, depth images can provide useful cues when combined with RGB images, but often perform poorly by themselves in semantic segmentation. In such a case, we do not consider adopting an expert network that only takes depth as input.

Let B represent the number of expert networks that get selected. We propose a *gating network*, denoted by $G(\mathbf{x})$, to decide which expert network should be activated. This gating network takes multimodal inputs \mathbf{x} to form a global view and then produces a B -dimensional sparse vector \mathbf{g} as output. The final output y takes the form of: $y = \sum_{i=1}^B g_i E_i(\mathbf{x}_i)$, where \mathbf{x}_i denotes the subset of modalities that the i -th expert takes as input.

Different from conventional MoEs [27] where the output is a weighted summation of expert networks and every branch is executed, in our formulation, the output of the gating network \mathbf{g} is a one-hot encoding, *i.e.*, only *one* branch is selected for each instance. Therefore, the computations

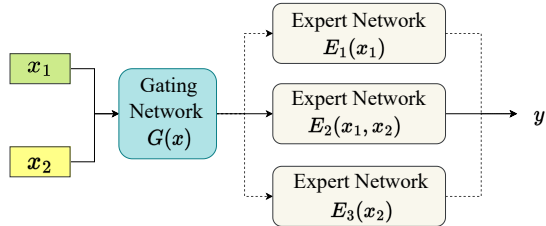


Figure 2. An illustration of modality-level DynMM, where input data has two modalities, denoted by x_1 and x_2 , and the output is denoted by y . We design a set of expert networks $\{E_i\}$ that specialize in different subsets of modalities and adopt a gating network $G(\mathbf{x})$ to generate data-dependent decisions on which expert network to select.

required for other expert networks can be saved. Note that since our expert network already covers a broad range of modality combinations, we only select one branch (as opposed to say selecting top K branches) during each forward pass for maximum computational savings. Figure 2 provides an illustration of the proposed design with 2 modalities and 3 expert networks (*i.e.*, $M = 2$ and $B = 3$).

The design of the gating network $G(\mathbf{x})$ follows two general requirements: (1) it should be computationally cheap to have a small overhead (2) it needs to be sufficiently expressive to make informative decisions on which expert to select. Various gating networks have been proposed previously; they are usually tailored for specific tasks and network architectures [11]. In the experiments, we consider different gating networks (*i.e.*, a multi-layer perceptron (MLP) gate, a transformer gate and a convolutional gate) for three multimodal tasks and provide the detailed description of our gating network architecture in Sec. 4.

One remaining problem is the training of gating network $G(\mathbf{x})$. Due to the non-differentiability of the discrete decisions given by $G(\mathbf{x})$, the network can not be directly trained with back-propagation. Thus, we propose reparameterization techniques and discuss them later in Sec. 3.4.

Finally, this gating network $G(\mathbf{x})$ is not restricted to taking input-level features; it can also take intermediate features per modality as inputs. Thus, modality-level DynMM can be plugged into any part of a multimodal network and achieve savings in computations after this gating network.

3.2. Fusion-level Decision

While the modality-level decisions directly impact the computational efficiency, completely skipping computations of one modality will likely lead to a downgraded performance for some challenging tasks, *e.g.*, semantic segmentation. Thus, we provide a finer-grain formulation of DynMM with fusion-level decisions next.

We first present the design of a *fusion cell*. Assume input data has M modalities, *i.e.*, $\mathbf{x} = (x_1, x_2, \dots, x_M)$.

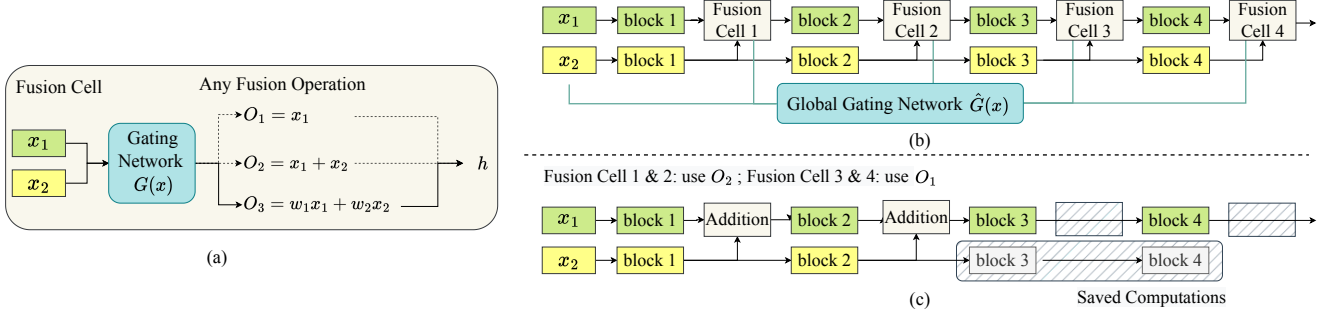


Figure 3. (a) An illustration of fusion-level DynMM, where input data has two modalities, denoted by x_1 and x_2 . We design a fusion cell with a set of candidate operations $\{O_i\}$ and a gating network $G(\mathbf{x})$. h represents output of the cell. (b) A dynamic multimodal architecture with stacked fusion cells, where we interlace static feature extraction blocks (colored with green and yellow) with dynamic fusion cells. Gating network $G(\mathbf{x})$ in four fusion cells are integrated as one global gating network $\hat{G}(\mathbf{x})$ that outputs decisions for four cells at once. (c) An example architecture when the gating network chooses O_2 for the first 2 fusion cells and O_1 for the last 2 cells. Consequently, computations of fusion cell 3 & 4 and feature extraction cell 3 & 4 for x_2 are saved.

Denote a set of fusion operations as $\{O_i\}$. O_i can be implemented as any function to fuse multimodal features, such as simple identity mapping (*i.e.*, $O_i = x_1$), addition (*i.e.*, $O_i = x_1 + x_2 + \dots + x_M$), concatenation (*i.e.*, $O_i = [x_1, x_2, \dots, x_M]$) and self-attention. Figure 3 (a) presents an example design of the fusion cell with two input modalities (*i.e.*, $\mathbf{x} = (x_1, x_2)$) and three operations (*i.e.*, $O_1 = x_1$, $O_2 = x_1 + x_2$, $O_3 = w_1x_1 + w_2x_2$), where w_1 and w_2 are learnable parameters. Note that here we simplify the operation set for illustration; in practice, we can always adopt more complex fusion operations in each cell to enlarge the representation power. Let B represent the total number of operations. A gating network $G(\mathbf{x})$ takes multimodal inputs and produces a B -dimensional vector \mathbf{g} that decides which operation to execute. The output of cell h can be represented as: $h = \sum_{i=1}^B g_i O_i(\mathbf{x})$. Following the previous discussion, we adopt hard gates (*i.e.*, \mathbf{g} is one-hot) for computational efficiency.

Fusion-level DynMM allows decisions at a finer granularity and in a more flexible way by stacking fusion cells to build a dynamic network. We provide an example architecture in Figure 3 (b) that we use in our experiments for semantic segmentation (x_1 and x_2 denote RGB and depth images, respectively). The network consists of four fusion blocks and a global gating network, which allows us to flexibly control the degree of fusion in a sample-wise manner. For instance, we show the resulting architecture in Figure 3 (c), when the gating network selects O_2 for fusion cell 1 & 2, and O_1 for fusion cell 3 & 4. This not only skips complex fusion operations that are not selected within the fusion cell, but also saves unnecessary computations in the feature extraction layer. Since we only adopt features from modality 1 after fusion cell 2, there is no need to further process features from modality 2. Thus, we can skip computations in the feature extraction layers for x_2 (*i.e.*, blocks

3-4 marked in gray). This strategy resembles early exiting in unimodal dynamic networks, yet with different motivations. In essence, fusion-level DynMM saves future fusion and modality-wise operations for some multimodal inputs when combining low-level features from each modality (*i.e.*, fusing at early stages) is sufficient for good predictions. On the other hand, for “hard” instances, DynMM provides the option of combining multimodal features in each cell with complex fusion operations for maximum representation power. Note that we replace the four individual gating networks $G(\mathbf{x})$ in each fusion cell with a global gating network $\hat{G}(\mathbf{x})$ for better integration; $\hat{G}(\mathbf{x})$ takes multimodal features (x_1, x_2) as input and makes decisions on which fusion operation to adopt for the four fusion cells.

This paradigm is especially helpful in tasks where the final prediction is mainly based on a dominant modality, while the other auxiliary modalities provide useful cues to improve the prediction. Fusion-level DynMM provides a flexible way to control *how* and *when* the auxiliary modality comes in to assist the main prediction process. *Progressive fusion* is achieved by our carefully designed fusion cell and dynamic architecture, leading to great computational savings, strong representation power and improved robustness.

Note that modality-level DynMM and fusion-level DynMM are two approaches targeting different granularity levels. In our experiments, we use modality-level DynMM to solve two classification tasks, while the fusion-level DynMM is used for the more challenging semantic segmentation task (*i.e.*, a dense prediction problem).

3.3. Training Objective

We notice that for both modality-level and fusion-level DynMM designs, the computation for each expert network E_i (operation O_i) is different. Normally, an expert network (an operation) that is computationally heavy has strong rep-

resentation power. If we directly train the network by minimizing a task-specific loss, the gating network is likely to learn a trivial solution that always chooses the branch with the heavy computation. To achieve efficient inference, we introduce a *resource-aware loss* function into the training objective. Let $C(E_i)$ denote the computation cost (e.g., MAdds) of executing an expert network E_i . Similarly, $C(O_{i,j})$ represents the computation cost of the i -th fusion operation in the j -th cell. Note that the computation cost can be pre-determined before training and is a constant term. The training objectives are shown below:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda \sum_{i=1}^B g_i C(E_i) \quad (\text{modality-level}) \quad (1)$$

$$\mathcal{L} = \mathcal{L}_{task} + \lambda \sum_{j=1}^F \sum_{i=1}^B g_i^{(j)} C(O_{i,j}) \quad (\text{fusion-level}) \quad (2)$$

where \mathcal{L}_{task} denotes the *task loss*, e.g., cross entropy between the network prediction and true label for classification. $\mathbf{g}^{(j)}$ represents the decision vector given by the j -th fusion cell. B is the total number of experts (operations) and F is the number of fusion cells. λ is a hyperparameter controlling the relative importance of the two loss terms.

The new objectives (1) and (2) account for the computation cost of executing each path and enables DynMM to achieve a desired tradeoff between accuracy and efficiency. We can adjust the value of λ based on the deployment constraints. For large λ , DynMM will prioritize lightweight computations for high computational efficiency. For small λ , DynMM will explore these computationally heavy paths more often, thus yielding higher accuracy.

3.4. Optimization

We aim to train DynMM in an end-to-end manner. Since the current gating network provides discrete decisions, the branch selection is not directly differentiable with respect to the gating network. Gumbel-softmax and reparameterization techniques [18] are introduced in the training process. Recall that \mathbf{g} denotes the desired one-hot B -dimensional decision vector produced by a gating network $G(\mathbf{x})$, i.e., $\mathbf{g} = \text{one-hot}(\arg \max_i G(\mathbf{x})_i)$. We adopt a real-valued soft vector $\tilde{\mathbf{g}}$ with the following form:

$$\tilde{g}_i = \frac{\exp((\log G(\mathbf{x})_i + b_i)/\tau)}{\sum_{j=1}^B \exp((\log G(\mathbf{x})_j + b_j)/\tau)} \quad i = 1, 2, \dots, B \quad (3)$$

where b_1, b_2, \dots, b_B are samples independently drawn from Gumbel(0, 1) [18] and τ denotes the softmax temperature. The distribution of $\tilde{\mathbf{g}}$ is more uniform with large τ and resembles a categorical distribution with small τ . $\tilde{\mathbf{g}}$ serves as a continuous, differentiable approximation of \mathbf{g} . We consider two training techniques: (a) Hard \mathbf{g} is replaced with

soft $\tilde{\mathbf{g}}$ in Equations (1)-(2) to enable back-propagation. During training, we anneal τ so that $\tilde{\mathbf{g}}$ gradually converges to a desired one-hot vector. (b) Following the straight-through technique [18], we adopt hard \mathbf{g} in the forward pass and soft $\tilde{\mathbf{g}}$ in the backward propagation with the gradient approximation $\nabla \mathbf{g} \approx \nabla \tilde{\mathbf{g}}$. In this way, the gating network still outputs a discrete decision during training. Note that we always use hard \mathbf{g} during inference for computational benefits. Next, we propose a two-stage training of DynMM that jointly optimizes the multimodal network and gating modules.

Stage I: Pre-training. We find that following sparse decisions of the gating network in the early stage of training can result in a biased optimization. Branches that are rarely selected have fewer and smaller weight updates; poor performance may result in them getting selected less often (thus never improving). The goal of a pre-training stage is to ensure that every branch of DynMM is fully optimized before the gating modules get involved. For modality-level DynMM, we sufficiently train each expert network at this stage. For fusion-level DynMM, we adopt random decisions (i.e., randomly an operation from the set of candidate operations) for each fusion cell so that each path of the dynamic network is optimized uniformly.

Stage II: Fine-tuning. We incorporate gating networks into our optimization process at this stage. With the reparameterization technique introduced above, we jointly optimize the dynamic network along with gating networks in an end-to-end fashion.

4. Experiments

4.1. Experimental Setup

We conduct experiments on three multimodal tasks: (a) movie genre classification on MM-IMDB [1]; (b) sentiment analysis on CMU-MOSEI [51]; (c) semantic segmentation on NYU Depth V2 [30]. To demonstrate the wide applicability of our proposed DynMM, we select the above three tasks that include different modalities (i.e., image and text in task (a), video, audio and text in task (b), RGB and depth images in task (c)). We adopt modality-level DynMM for the first two tasks and fusion-level DynMM for the more challenging semantic segmentation task. Due to space limitations, we present: (1) implementation details; (2) visualization of the gating network decision; (3) an analysis of varying regularization strength λ ; and (4) an ablation study on training strategies of DynMM in the Appendix.

4.2. Movie Genre Classification

MM-IMDB is the largest publicly available multimodal dataset for genre prediction on movies. It comprises 25,959 movie titles, metadata and movie posters. We select two movie genres (i.e., drama and comedy) for multi-label classification from posters (image modality) and text descrip-

Method	Modality	Micro F1 (%)	Macro F1 (%)	MAdds (M)
Image Network	I	39.99	25.26	5.0
Text Network (E_1)	T	59.16	47.21	0.7
Late Fusion [24] (E_2)		59.55	50.94	10.3
LRTF [26]		59.18	49.26	10.3
MI-Matrix [19]		58.45	48.36	10.3
DynMM-a		59.57	48.84	1.6
DynMM-b		59.59	50.42	7.8
DynMM-c	I+T	59.72	51.20	9.8
DynMM-d		60.35	51.60	12.1

Table 1. Results on the MM-IMDB Movie Genre Classification. Modalities I and T denote image and text, respectively. The computation cost is measured by multiply-add operations (MAdds) with one image-text pair as the input. M denotes million. Each DynMM variant is obtained using a different value of the regularization hyperparameter λ during training.

tions (text modality). We follow the original data split in [1], and use 15,552 data for training, 2,608 for validation and 7,799 for testing. For preprocessing, we adopt the same method as [1, 24] to extract text and image features.

We adopt two expert networks for this task, namely, a unimodal network E_1 that takes textual features as input and another multimodal network E_2 that adopts late fusion [24] to combine image and text features. We do not consider the use of an image-only network here due to its poor performance on this task. The gating network is a 2-layer MLP with hidden dimension of 128, which takes concatenated image and text features as input and outputs a 2-dimensional vector for expert network selection. We set the temperature of Gumbel-softmax as 1 and adopt straight-through training (*i.e.*, the gating network outputs a one-hot decision vector in the forward propagation).

Table 1 provides the comparison of our proposed modality-level DynMM with static unimodal networks and multimodal networks. We provide results of DynMM under different resource requirements (*i.e.*, use different λ in the loss). From Table 1, we can see that DynMM achieves a good balance between computational efficiency and performance. Compared to the static E_2 network, DynMM-c improves both MAdds and macro F1 score. DynMM-d provides maximum representation power by using soft gates (which leads to more computation) and achieves best micro and macro F1 scores. On the other hand, DynMM-a involves much less computation, while still maintaining good performance (outperforms E_1 by 1.6% in macro F1). This demonstrates the great flexibility and efficacy of DynMM.

In addition, we vary λ in Equation (1) to control the importance of resource loss during training. The resulting DynMM models have varying computation costs and performance, as shown in Figure 4 (a). On one hand, when

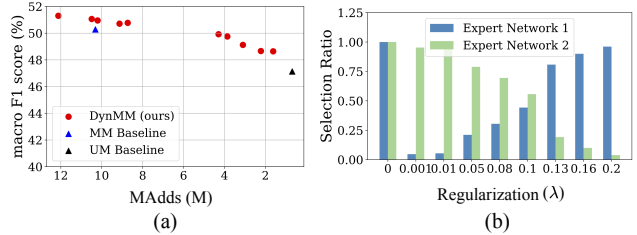


Figure 4. Analysis of DynMM with varying resource regularization strength (λ) on MM-IMDB. (a): Comparison of DynMM with static unimodal (UM) and multimodal (MM) baselines. (b): Branch selection ratio in DynMM with respect to λ . DynMM offers a wide range of choices that balance computation and learning behavior well.

compared against a multimodal baseline that is computationally heavy, DynMM maintains good performance with much fewer MAdds. On the other hand, DynMM has better representation power than a unimodal network and thus improves the F1 score. Figure 4 (b) shows the selection ratio of each expert network in DynMM with respect to λ . We observe that as λ increases, DynMM focuses more on reducing computation and thus is more likely to select expert network 1 (E_1) with a small computation cost. Note that for the $\lambda = 0$ case, we adopt soft gates, *i.e.*, every expert network is activated and the output is a weighted combination of predictions given by the two expert networks. Thus, DynMM achieves the best performance at the cost of increased computation. This also demonstrates the flexibility of DynMM, as we can easily adjust λ to target high performance or high inference efficiency.

4.3. Sentiment Analysis

CMU Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) is the largest dataset of sentiment analysis and emotion recognition. It contains 3,228 real-world online videos from more than 1000 speakers and 250 topics. Each video is split into short segments of 10-20 seconds. Each segment is annotated for a sentiment from -3 (strongly negative) to 3 (strongly positive). The task is to predict the sentiment scores from video, audio and text. Following [24], we use 16,265 data for training, 1,869 data for validation and 4,643 data for testing. The feature extraction steps are the same as [24].

As text is the best performing modality in this task, we adopt a unimodal network that takes textual features as input to be the expert network E_1 . The second expert network (E_2) of our DynMM is selected as a late fusion network [24] that receives inputs from three modalities. The gating network is designed as a lightweight transformer network with hidden dimension equal to 512 and 2 attention heads, followed by a linear layer. The gating network receives concatenated features from three modalities and

Method	Modality	Acc ² (%)	MAE	MAdds (M)
Video Network	V	69.02	0.80	123.1
Audio Network	A	67.68	0.82	123.3
Text Network (E_1)	T	78.35	0.62	124.7
Early Fusion [24]	V+A+T	78.45	0.65	313.5
Late Fusion [24] (E_2)		79.54	0.60	309.6
DynMM-a	V+A+T	79.07	0.62	165.5
DynMM-b		79.73	0.61	254.5
DynMM-c		79.75	0.60	295.8

Table 2. Results on CMU-MOSEI Sentiment Analysis. Modalities V, A, T represent video, audio and text, respectively. Acc² denotes binary accuracy (*i.e.*, positive/negative sentiments) and MAE represents mean absolute error. MAdds are measured with a video-audio-text tuple. Each DynMM variant is obtained using a different value of the regularization hyperparameter λ during training.

- Hi, I'm here to review *In the Name of the King*
- It's a film currently in theaters
- (uhh) It's based on the xxx series of video games
- This one was horrible
...
- (uhh) It tries to be *Lord of the Rings*
- It has a many of orge like creatures, a wizard, a medieval time setting
- Basically stole everything out of *Lord of the Rings* but made a million times worse
- (umhh) It's just everyone stay away from this film
- It was a horrible movie
- (umm) So yea that's *In the Name of the King* in a nutshell



Figure 5. We visualize a few test instances on CMU-MOSEI for a negative sentiment. DynMM identifies sentences marked with red as “easy” instances and only uses textual information for prediction. For sentences marked with blue, DynMM takes multimodal inputs (*i.e.*, video+audio+text) for more accurate predictions.

generates sample-wise decisions on which expert network to activate during inference time. We set temperature of Gumbel-softmax as 1 and adopt straight-through training.

Results are summarized in Table 2. We provide three DynMM networks trained with different λ . Compared with the best performing static network (*i.e.*, Late Fusion), DynMM-a can reduce computations by 46.5% with a slightly decreased accuracy (*i.e.*, -0.47%). By allowing more computation, DynMM-b improves both inference efficiency (*i.e.*, reduce MAdds by 17.8%) and prediction accuracy. Finally, DynMM-c further improves the accuracy by trading off some computation; it achieves best accuracy and smallest mean absolute error with reduced computation cost. These results demonstrate the great advantages of dynamic multimodal fusion. Since multimodal data naturally brings redundancy, we observe that many computations can be reduced without loss in accuracy.

To have an intuitive sense of our gating network deci-

sion on which modality to select, we provide visualization results of several test instances in Figure 5. For simplicity only the text modality is shown here, and the other two modalities (*i.e.*, video and audio) are omitted. The gating network chooses E_1 for sentences marked with red and E_2 for sentences marked with dark blue. We find that the sentences marked with red often possess strong evidence indicating the sentiments of this sample, *e.g.*, ‘horrible’, ‘amazingly good’. Therefore, they belong to the “easy” samples category that can be correctly predicted using the text modality alone. On the contrary, the sentences marked with dark blue are vague and require additional modalities to help with the prediction. These results indicate that the gating function is well trained and can provide reasonable decisions based on input characteristics.

4.4. Semantic Segmentation

NYU Depth V2 is an indoor dataset for semantic segmentation. It contains 1,449 RGB-D images with 40-class labels; 795 images are used for training and 654 images are for testing. The two modalities are RGB and depth images.

Method	mIoU (%)	Depth Enc MAdds (G)	MAdds Reduction (%)
ESANet [35] (baseline)	50.5	24.7	-
DynMM (Stage I)	48.5	11.7	52.6%
DynMM-a (Stage II)	49.9	11.1	55.1%
DynMM-b (Stage II)	51.0	19.5	21.1%

Table 3. Results on RGB-D semantic segmentation. mIoU denotes mean Intersection-over-Union. MAdds are calculated for input size of $3 \times 480 \times 640$. G stands for Giga.

We adopt fusion-level DynMM for this task and base our dynamic architecture design on a (static) efficient architecture, ESANet [35]. As illustrated in Figure 3, we incorporate four fusion cells in the encoder design, where each fusion cell contains two operations. Operation 1 is an identity mapping of RGB features, *i.e.*, $O_1 = x_1$. For the second operation, we use channel attention fusion, where features from both modalities are first reweighted with a Squeeze and Excitation module [15] and then added element-wisely. Two ResNet-50 [13] are used as feature extraction models for RGB and depth modality. The decoder design is identical to [35]. The gating network comprises a pipeline of 2 convolution blocks with kernel size 5×5 and stride size 2, a global average pooling and a linear layer. RGB and depth features after the first convolutional layer are concatenated together and passed to the convolutional gate. The gating network outputs a 4-dimensional vector per sample that determines which operation to select for each fusion cell. We experiment with two training strategies: (1) DynMM-a in Table 3 is trained with straight-through technique with Gumbel-softmax temperature $\tau = 1$; (2) We

Method	Modality	Backbone	mIoU (%)	MAdds (G)
LW-RefineNet [31]	RGB	ResNet-50	41.7	38.5
LW-RefineNet [31]		ResNet-101	43.6	61.2
ACNet [16]	RGB+D	ResNet-50	48.3	126.2
SA-Gate [8]		ResNet-50	50.4	147.6
CEN [45]		ResNet-101	51.1	618.3
ESANet [35]		ResNet-50	50.5	56.9
DynMM-a	RGB+D	ResNet-50	49.9	43.4
DynMM-b		ResNet-50	51.0	52.2

Table 4. Comparison of our approach with SOTA methods for RGB-D semantic segmentation on NYU Depth V2 test data.

obtain DynMM-b in Table 3 by exponentially decaying τ from 1 to 0.0001 during 500 epochs.

Table 3 provides the detailed results of fusion-level DynMM. We report performance of DynMM after first-stage training in the second row; its great performance validates the design of our random gating function in the pre-training stage. This also lends support to our claim that there exists a lot of redundancy in multimodal networks. Utilizing the finding that depth modality plays an auxiliary role in this task, fusion-level DynMM effectively reduces computations of the depth encoder. DynMM-a reduces MAdds by 55.1% with only -0.4% mIoU drop. Furthermore, DynMM-b achieves a mIoU improvement of 0.7% and 21.1% reduction in MAdds at the same time, thus demonstrating the superiority of DynMM over static fusion.

Table 4 presents a comparison of the resulting DynMM-a and DynMM-b with SOTA semantic segmentation methods. For baseline methods, we list mIoU reported in their original papers and report MAdds. These results clearly show that our proposed method achieves the best balance between performance and efficiency. The computation cost of DynMM is similar to a unimodal lightweight RefineNet, yet its performance can match methods that use ResNet-101 as the backbone and involve significantly larger MAdds.

Finally, we conduct experiments to demonstrate the improved robustness of DynMM compared to ESANet. We consider three settings by injecting random Gaussian noise with probability 1/3 to (1) RGB modality; (2) depth modality and (3) both modalities. We experiment with different degrees of random Gaussian noise and plot the performance degradation of two approaches in Figure 6. From the figure, we observe that the performance gap between DynMM and ESANet becomes larger when the noise level of depth images increases; This demonstrates another advantage of DynMM in reducing data noise and improving robustness. Figure 7 shows some qualitative segmentation results. While ESANet generates reasonable predictions in the normal setting (*i.e.*, first and third row), its performance becomes significantly worse when multimodal data is perturbed by noise (*i.e.*, the second and fourth row). On the contrary, our DynMM is robust to noise and provides a good

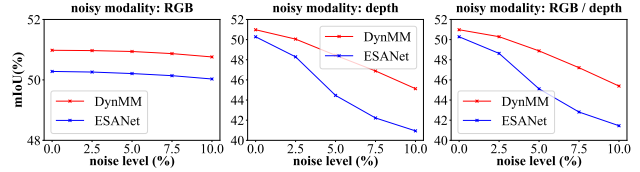


Figure 6. DynMM vs. ESANet on NYU Depth V2 with different degrees of Gaussian noise injected into RGB / depth images.

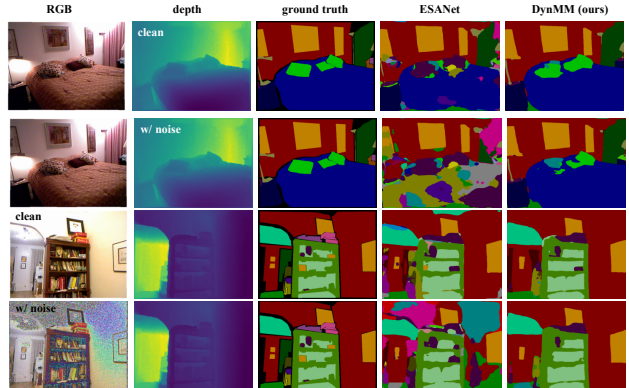


Figure 7. Qualitative segmentation results on NYU Depth V2. DynMM is more robust to noisy multimodal data compared with the static ESANet.

prediction for both scenarios. These results suggest the potential of a dynamic neural network architecture for improving robustness of multimodal fusion.

5. Conclusion

Multimodal data enable models to learn from an enriched representation space, but it also bring significant redundancy. Motivated by this observation, we have proposed dynamic multimodal fusion (DynMM), a new approach that adaptively fuses inputs during inference. Experimental results on three very different multimodal tasks demonstrate the efficacy of DynMM. More importantly, our work demonstrates the potential of dynamic multimodal fusion and opens up a new research direction. Considering the benefits of a dynamic architecture (*i.e.*, reduced computation, improved performance and robustness), we believe that developing dynamic networks tailored for multimodal fusion is a topic worthy of further investigations.

DynMM has limitations that we plan to address through three areas of improvement in our future work. These include designing better dynamic architectures that can account for multimodal redundancy, extending DynMM to sequential decision-making tasks, such as long video prediction and exploring the performance of DynMM on different multimodal tasks and modalities.

References

- [1] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017. 5, 6
- [2] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018. 1
- [3] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015. 2
- [4] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, volume 70, pages 527–536. PMLR, 2017. 2
- [5] Shaofeng Cai, Yao Shu, and Wei Wang. Dynamic routing networks. In *Winter Conference on Applications of Computer Vision*, pages 3588–3597, 2021. 2
- [6] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Udmhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *International Conference on Image Processing*, pages 168–172. IEEE, 2015. 1, 2
- [7] Chunlin Chen and Qiang Ling. Adaptive convolution for object detection. *IEEE Transactions on Multimedia*, 21(12):3205–3217, 2019. 2
- [8] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *European Conference on Computer Vision*, pages 561–577. Springer, 2020. 2, 8
- [9] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Conference on Computer Vision and Pattern Recognition*, pages 10457–10467, 2020. 2, 3
- [10] Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. 2
- [11] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 3
- [12] Zongbo Han, Fan Yang, Junzhou Huang, Changqing Zhang, and Jianhua Yao. Multimodal dynamics: Dynamical fusion for trustworthy multimodal classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20707–20717, 2022. 2, 3
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7
- [14] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *International Conference on Computer Vision*, pages 4193–4202, 2017. 1
- [15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 7
- [16] Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. Acnet: Attention based network to exploit complementary features for rgb-d semantic segmentation. In *International Conference on Image Processing*, pages 1440–1444. IEEE, 2019. 8
- [17] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 2752–2761, 2018. 2
- [18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 5
- [19] Siddhant M Jayakumar, Wojciech M Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. Multiplicative interactions and where to find them. 2020. 6
- [20] Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L Iuzolino, and Kazuhito Koishida. Mmtm: Multimodal transfer module for cnn fusion. In *Conference on Computer Vision and Pattern Recognition*, pages 13289–13299, 2020. 1, 2
- [21] Ramandeep Kaur and Sandeep Kautish. Multimodal sentiment analysis: A survey and comparison. *International Journal of Service Science, Management, Engineering, and Technology*, 10(2):38–58, 2019. 1, 2
- [22] Michelle A Lee, Matthew Tan, Yuke Zhu, and Jeannette Bohg. Detect, reject, correct: Crossmodal compensation of corrupted sensors. In *International Conference on Robotics and Automation*, pages 909–916. IEEE, 2021. 2
- [23] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 8553–8562, 2020. 2
- [24] Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A Lee, Yuke Zhu, et al. Multibench: Multiscale benchmarks for multimodal representation learning. *arXiv preprint arXiv:2107.07502*, 2021. 6, 7
- [25] Kuan Liu, Yanen Li, Ning Xu, and Prem Natarajan. Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730*, 2018. 1, 2
- [26] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064*, 2018. 1, 2, 6
- [27] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014. 3
- [28] Ravi Teja Mullaipudi, William R Mark, Noam Shazeer, and Kayvon Fatahalian. Hydranets: Specialized dynamic architectures for efficient inference. In *Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2018. 2
- [29] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for

- multimodal fusion. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2
- [30] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *European Conference on Computer Vision*, 2012. 5
- [31] Vladimir Nekrasov, Chunhua Shen, and Ian Reid. Light-weight refinenet for real-time semantic segmentation. *arXiv preprint arXiv:1810.03272*, 2018. 8
- [32] Rameswar Panda, Chun-Fu Richard Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris. Adamml: Adaptive multi-modal learning for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7576–7585, 2021. 2, 3
- [33] Juan-Manuel Pérez-Rúa, Valentin Vielzeuf, Stéphane Pateux, Moez Baccouche, and Frédéric Jurie. Mfas: Multimodal fusion architecture search. In *Conference on Computer Vision and Pattern Recognition*, pages 6966–6975, 2019. 1, 2
- [34] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in Neural Information Processing Systems*, 30, 2017. 2
- [35] Daniel Seichter, Mona Köhler, Benjamin Lewandowski, Tim Wengefeld, and Horst-Michael Gross. Efficient rgb-d semantic segmentation for indoor scene analysis. In *International Conference on Robotics and Automation*, pages 13525–13531. IEEE, 2021. 1, 2, 7, 8, 11
- [36] Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1045–1058, 2017. 1, 2
- [37] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 2
- [38] Mohammad Soleymani, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14, 2017. 1, 2
- [39] Peng Sun, Wenhui Zhang, Huanyu Wang, Songyuan Li, and Xi Li. Deep rgb-d saliency detection with depth-sensitive attention and automatic multi-modal fusion. In *Conference on Computer Vision and Pattern Recognition*, pages 1407–1417, 2021. 1, 2
- [40] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, pages 2464–2469. IEEE, 2016. 2
- [41] Hiroki Tokunaga, Yuki Teramoto, Akihiko Yoshizawa, and Ryoma Bise. Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology. In *Conference on Computer Vision and Pattern Recognition*, pages 12597–12606, 2019. 2
- [42] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. Centralnet: a multilayer approach for multimodal fusion. In *European Conference on Computer Vision Workshops*, pages 0–0, 2018. 1
- [43] Valentin Vielzeuf, Stéphane Pateux, and Frédéric Jurie. Temporal multimodal fusion for video emotion classification in the wild. In *International Conference on Multimodal Interaction*, pages 569–576, 2017. 1, 2
- [44] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, pages 409–424, 2018. 2
- [45] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. Deep multimodal fusion by channel exchanging. *Advances in Neural Information Processing Systems*, 33:4835–4845, 2020. 1, 2, 8
- [46] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. *Advances in Neural Information Processing Systems*, 33:2432–2444, 2020. 2
- [47] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in Neural Information Processing Systems*, 32, 2019. 2
- [48] Mingyu Yang, Yu Chen, and Hun-Seok Kim. Efficient deep visual and inertial odometry with adaptive visual modality selection. *arXiv preprint arXiv:2205.06187*, 2022. 2
- [49] Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. Deep multimodal neural architecture search. In *International Conference on Multimedia*, pages 3743–3752, 2020. 1, 2
- [50] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*, 2017. 1, 2
- [51] Amir Zadeh and Paul Pu. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Annual Meeting of the Association for Computational Linguistics*, 2018. 1, 5
- [52] Hong-Yu Zhou, Bin-Bin Gao, and Jianxin Wu. Adaptive feeding: Achieving fast and accurate detections by adaptively combining object detectors. In *International Conference on Computer Vision*, pages 3505–3513, 2017. 2

In this appendix, we present: (1) implementation details; (2) visualization results of decisions given by the gating network (on NYU Depth V2); (3) an analysis of varying regularization strength λ (on CMU-MOSEI); and (4) an ablation study on proposed training strategies (on NYU Depth V2).

A. Implementation Details

MM-IMDB. E_1 is a unimodal text network with 2-layer MLPs (hidden dimension=512) as the text encoder and the decoder. E_2 is a multimodal late fusion network, where we use the text and image encoders to extract features, concatenate the unimodal features and then pass the concatenated features to a MLP decoder (hidden dimension=1024). The text encoder is the same as in E_1 and the image encoder is a 2-layer MLP (hidden dimension=1024). We use AdamW optimizer with lr=1e-4 and weight decay=1e-2.

CMU-MOSEI. E_1 is a text network consisting of a 5-layer transformer encoder (hidden dimension=120; 5 attention heads) and a 2-layer MLP decoder (hidden dimension=64). E_2 is a multimodal late fusion network with video, audio, and text encoders being 5-layer transformers and a 2-layer MLP decoder (hidden dimension=128). We use AdamW optimizer with lr=1e-4 and weight decay=1e-4.

NYU Depth V2. The image and depth encoder is a ResNet-50 and the decoder is the same as in ESANet [35]. We use SGD optimizer with weight decay=1e-4 and momentum=0.9, also OneCycleLR with max_lr=1e-2.

The gating networks are designed to match the E_1 and E_2 model architectures. Therefore, we use a MLP gate for MM-IMDB, a transformer gate for CMU-MOSEI and a convolution gate for NYU Depth V2.

$C(E_i)$ in Equations (1)-(2) is set as the MACs required to do one forward pass with E_i . Take MM-IMDB for example: the MACs for executing E_1 and E_2 are 1.25M and 10.87M, respectively. The resource loss of one data sample is λ if the gating network selects E_1 and $\lambda \times \frac{10.87}{1.25}$ if E_2 is selected. The DynMM variants reported in Table 1-2 are obtained using different values of the regularization parameter λ .

B. Visualization Results

In our proposed DynMM, the gating network is crucial as it provides data-dependent decisions on which expert network to adopt. For modality-level DynMM, we have provided visualization of the gating network decisions for some test instances on CMU-MOSEI in Figure 5 in the main paper. Similarly, for fusion-level DynMM, we visualize several test instances on NYU Depth V2 and the resulting architecture in Figure 8 in the Appendix.

From Figure 8, we can see that DynMM adaptively executes the forward path for multimodal inputs. The depth

features are combined with the RGB features to different degrees, determined by the gating network in DynMM. This provides a flexible way to control multimodal fusion in a sample-wise manner. For the RGB-D images in the upper figure, DynMM performs one-time fusion for multimodal features after the first block and saves computations of depth blocks 2-4. For the more challenging test samples in the lower figure, DynMM decides to fuse features in every layer to better incorporate multimodal information. Due to the dynamic architecture, DynMM achieves a good balance between efficiency and performance.

C. Analysis of Regularization Strength

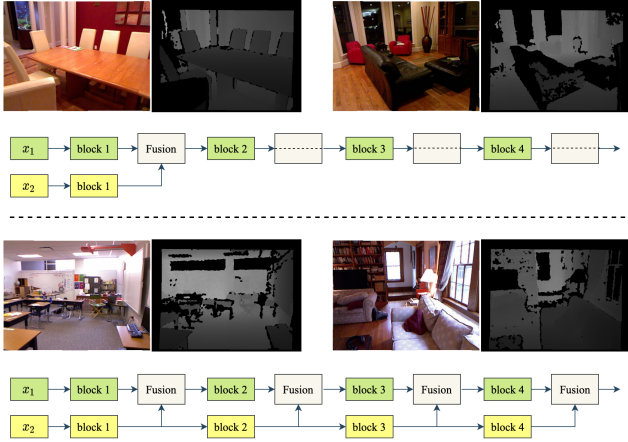
Recall that we propose a resource-aware loss function in Equation (1) and (2) in the main paper, where λ is a hyperparameter controlling the relative importance of task loss and computation cost loss. Similar to Figure 4 in the main paper (*i.e.*, an analysis of λ on MM-IMDB), we vary λ when training DynMM on CMU-MOSEI sentiment analysis and report its computation cost and performance corresponding to each λ value. The results are provided in Figure 9 in this Appendix. From Figure 9 (a), we can see that DynMM achieves a good balance between inference efficiency and accuracy. Moreover, DynMM offers a wide range of choices that can be controlled by λ , thus showing great flexibility. Figure 9 (b) shows the branch selection ratio of DynMM for different λ . When λ is small, DynMM focuses more on performance and chooses expert network 2 most of the time. As λ increases, more test samples are routed to the expert network 1 that requires fewer computations.

D. Ablation Study

To verify the efficacy of our proposed training strategies, we present an ablation study of RGB-D semantic segmentation on the NYU Depth V2 data. We train DynMM under three settings: (1) We omit the pre-training stage and train DynMM in one stage. (2) In the second stage of training, we freeze the weights of the multimodal architecture and only fine-tune the gating network. (3) We adopt our proposed two-stage training with joint optimization of the multimodal

Method	Two-stage Training	Joint Optimization	mIoU (%)
Baseline			50.3
		✓	49.2
DynMM	✓		50.2
	✓	✓	51.0

Table 5. Ablation study on RGB-D semantic segmentation. Baseline refers to a static model (ESANet).



joint optimization achieves the overall best performance.

Figure 8. We visualize a few test instances on the NYU Depth V2 data. x_1 and x_2 denote RGB and depth images, respectively. The corresponding network architecture based on the gating network decision is shown. The upper figure shows examples when the gating network chooses an early fusion architecture. DynMM skips computations of the depth extraction layers, thus achieving inference savings. The lower figure shows examples when the gating network decides to fuse representations at every middle layer.

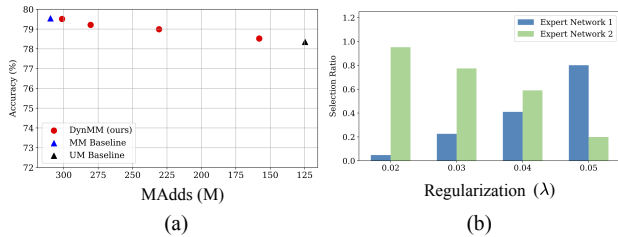


Figure 9. Analysis of DynMM with varying resource regularization strength (λ) on CMU-MOSEI. (a): comparison of DynMM with static unimodal (UM) and multimodal (MM) baselines. (b): branch selection ratio in DynMM with respect to λ .

network and gating network. The other training parameters (e.g., learning rate, resource regularization strength λ) are identical. The results are shown in Table 5 below.

Table 5 demonstrates the advantages of our proposed training strategies. We observe that DynMM with one-stage training does not have a dynamic architecture, i.e., all test samples are routed to one particular forward path. Without a pre-training stage, every forward path is not equally optimized. Biased optimization further leads to suboptimal performance (i.e., an mIoU of 49.2%). Apart from two-stage training, joint optimization also plays an important role. We observe a +0.8% mIoU improvement with end-to-end training. The possible reason is that (static) feature extraction layers also improve in the joint optimization process; they provide more informative features as input to the gating network to a better gating network decision. Therefore, the