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- 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

¹Our code is available at https://github.com/zihuiXue/DynMM.
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 $x = (x_1, x_2, \ldots, x_M)$. 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 $G(x)$, to decide which expert network should be activated. This gating network takes multimodal inputs $x$ to form a global view and then produces a $B$-dimensional sparse vector $g$ as output. The final output $y$ takes the form of: $y = \sum_{i=1}^{B} y_i E_i(x_i)$, where $x_i$ 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 $g$ is a one-hot encoding, i.e., only one branch is selected for each instance. Therefore, the computations 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 $G(x)$ 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 $G(x)$. Due to the non-differentiability of the discrete decisions given by $G(x)$, the network cannot be directly trained with back-propagation. Thus, we propose reparameterization techniques and discuss them later in Sec. 3.4.

Finally, this gating network $G(x)$ 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., $x = (x_1, x_2, \ldots, x_M)$.

![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 $G(x)$ to generate data-dependent decisions on which expert network to select.](image-url)
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., \(O_i = x_1 + x_2 + \cdots + x_M\)), concatenation (i.e., \(O_i = [x_1, x_2, \cdots, x_M]\)) and self-attention. Figure 3 (a) presents an example design of the fusion cell with two input modalities (i.e., \(x = (x_1, x_2)\)) and three operations (i.e., \(O_1 = x_1, O_2 = x_1 + x_2, O_3 = w_1 x_1 + w_2 x_2\)), 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 \(G(x)\) takes multimodal inputs and produces a \(B\)-dimensional vector \(g\) that decides which operation to execute. The output of cell \(h\) can be represented as: \(h = \sum_{i=1}^{B} g_i O_i(x)\). Following the previous discussion, we adopt hard gates (i.e., \(g\) 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 \(G(x)\) in each fusion cell with a global gating network \(\hat{G}(x)\) for better integration; \(\hat{G}(x)\) 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).
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:

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

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

where $L_{task}$ denotes the task loss, e.g., cross entropy between the network prediction and true label for classification. $g_{i}^{(j)}$ 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. $\lambda$ 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 $\lambda$ based on the deployment constraints. For large $\lambda$, DynMM will prioritize lightweight computations for high computational efficiency. For small $\lambda$, 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 $g$ denotes the desired one-hot $B$-dimensional decision vector produced by a gating network $G(x)$, i.e., $g = \text{one-hot}(\arg\max_{i} G(x)_i)$. We adopt a real-valued soft vector $\tilde{g}$ with the following form:

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

where $b_1, b_2, \ldots, b_B$ are samples independently drawn from $\text{Gumbel}(0, 1)$ [18] and $\tau$ denotes the softmax temperature. The distribution of $\tilde{g}$ is more uniform with large $\tau$ and resembles a categorical distribution with small $\tau$. $\tilde{g}$ serves as a continuous, differentiable approximation of $g$. We consider two training techniques: (a) Hard $g$ is replaced with soft $\tilde{g}$ in Equations (1)-(2) to enable back-propagation. During training, we anneal $\tau$ so that $\tilde{g}$ gradually converges to a desired one-hot vector. (b) Following the straight-through technique [18], we adopt hard $g$ in the forward pass and soft $\tilde{g}$ in the backward propagation with the gradient approximation $\nabla g \approx \nabla \tilde{g}$. In this way, the gating network still outputs a discrete decision during training. Note that we always use hard $g$ 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 $\lambda$; 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-
As shown in Figure 4, DynMM achieves the best performance at the cost of increased computation. This also demonstrates the flexibility of DynMM, as we can easily adjust \( \lambda \) 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 processes them using a 2-layer MLP with hidden dimension equal to 512.

Table 1 provides the comparison of our proposed modality-level DynMM with static unimodal and multimodal networks. We provide results of DynMM under different resource requirements (i.e., use different \( \lambda \) 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 \( \lambda \) 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 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 \( \lambda \). We observe that as \( \lambda \) 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 \( \lambda \) to target high performance or high inference efficiency.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Micro F1 (%)</th>
<th>Macro F1 (%)</th>
<th>MAdds (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Network</td>
<td>I</td>
<td>39.99</td>
<td>25.26</td>
<td>5.0</td>
</tr>
<tr>
<td>Text Network (( E_1 ))</td>
<td>T</td>
<td>59.16</td>
<td>47.21</td>
<td>0.7</td>
</tr>
<tr>
<td>Late Fusion [23] (( E_2 ))</td>
<td>I+T</td>
<td>59.55</td>
<td>50.94</td>
<td>10.3</td>
</tr>
<tr>
<td>LRTF [26]</td>
<td></td>
<td>59.18</td>
<td>49.26</td>
<td>10.3</td>
</tr>
<tr>
<td>MI-Matrix [19]</td>
<td></td>
<td>58.45</td>
<td>48.36</td>
<td>10.3</td>
</tr>
</tbody>
</table>

DynMM-a: 59.57 48.84 1.6
DynMM-b: 59.59 50.42 7.8
DynMM-c: 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 \( \lambda \) during training.

Figure 4. Analysis of DynMM with varying resource regularization strength (\( \lambda \)) on MM-IMDB. (a): Comparison of DynMM with static unimodal (UM) and multimodal (MM) baselines. (b): Branch selection ratio in DynMM with respect to \( \lambda \). DynMM offers a wide range of choices that balance computation and learning behavior well.
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 $\lambda$. Compared with the best 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 efficiency (i.e., -0.47%).

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.

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 $5 \times 5$ 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

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Acc$^2$ (%)</th>
<th>MAE</th>
<th>MAdds (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Network</td>
<td>V</td>
<td>69.02</td>
<td>0.80</td>
<td>123.1</td>
</tr>
<tr>
<td>Audio Network</td>
<td>A</td>
<td>67.68</td>
<td>0.82</td>
<td>123.3</td>
</tr>
<tr>
<td>Text Network ($E_1$)</td>
<td>T</td>
<td>78.35</td>
<td>0.62</td>
<td>124.7</td>
</tr>
<tr>
<td>Early Fusion [24]</td>
<td>V+A+T</td>
<td>79.73</td>
<td>0.61</td>
<td>254.5</td>
</tr>
<tr>
<td>Late Fusion [24] ($E_2$)</td>
<td>V+A+T</td>
<td>79.73</td>
<td>0.60</td>
<td>298.8</td>
</tr>
</tbody>
</table>

Table 2. Results on CMU-MOSEI Sentiment Analysis. Modalities V, A, T represent video, audio and text, respectively. Acc$^2$ 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 $\lambda$ during training.
obtain DynMM-b in Table 3 by exponentially decaying $\tau$ 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 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Backbone</th>
<th>mIoU (%)</th>
<th>MAdds (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LW-RefineNet</td>
<td>RGB</td>
<td>ResNet-50</td>
<td>41.7</td>
<td>38.5</td>
</tr>
<tr>
<td>LW-RefineNet</td>
<td>RGB</td>
<td>ResNet-101</td>
<td>43.6</td>
<td>61.2</td>
</tr>
<tr>
<td>ACNet [16]</td>
<td>RGB</td>
<td>ResNet50</td>
<td>48.3</td>
<td>126.2</td>
</tr>
<tr>
<td>SA-Gate [8]</td>
<td>RGB+D</td>
<td>ResNet-50</td>
<td>50.4</td>
<td>147.6</td>
</tr>
<tr>
<td>ESANet [35]</td>
<td>RGB+D</td>
<td>ResNet-50</td>
<td>50.5</td>
<td>56.9</td>
</tr>
</tbody>
</table>

DynMM-a     RGB+D  ResNet-50  49.9     43.4
DynMM-b     RGB+D  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.
References


multimodal fusion. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2


