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 concantenated 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).

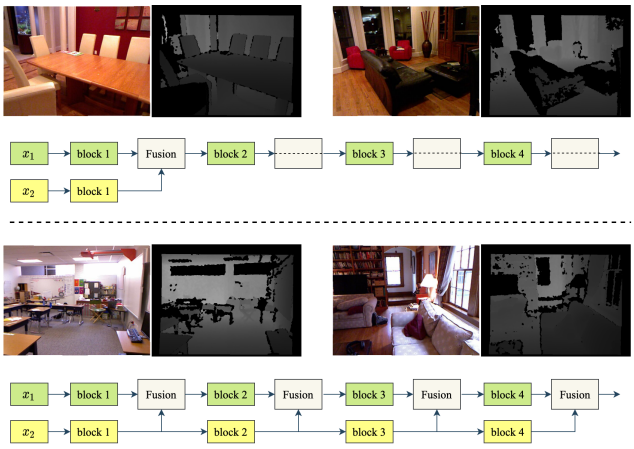

Figure 8. We visualize a few test instances on the NYU Depth V2 data. x_1 and x_2 denote RGB and depth images, respectively. The corresponding network architecture based on the gating network decision is shown. The upper figure shows examples when the gating network chooses an early fusion architecture. 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.

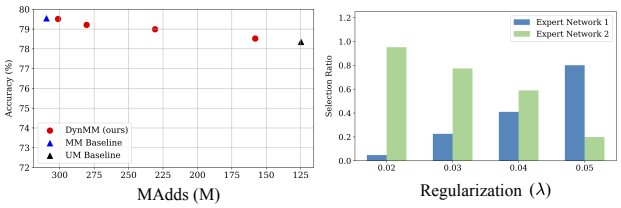

Figure 9. Analysis of DynMM with varying resource regularization strength (λ) on CMU-MOSEI. (a): comparison of DynMM with static unimodal (UM) and multimodal (MM) baselines. (b): branch selection ratio in DynMM with respect to λ.

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 joint optimization achieves the overall best performance.