Sparse MoE as a New Treatment: Addressing Forgetting, Fitting, Learning Issues in Multi-Modal Multi-Task Learning

Jie Peng[1], Sukwon Yun[2], Kaixiong Zhou[3], Ruida Zhou[4], Thomas Hartvigsen[5], Yanyong Zhang[1], Zhangyang Wang[6], Tianlong Chen[2]
[1]University of Science and Technology of China, [2]University of North Carolina at Chapel Hill
[3]Massachusetts Institute of Technology, [4]University of California, Los Angeles
[5]University of Virginia, [6]University of Texas at Austin
pengjieb@mail.ustc.edu.cn, {swyun, tianlong}@cs.unc.edu, Kz34@mit.edu, ruida@g.ucla.edu, tomh@mit.edu, yanyongz@ustc.edu.cn, atlaswang@utexas.edu

Sparse Mixture-of-Experts (SMoE) is a promising paradigm that can be easily tailored for multi-task learning. Its conditional computing nature allows us to organically allocate relevant parts of a model for performant and efficient predictions. However, several under-explored pain points persist, especially when considering scenarios with both multiple modalities and tasks: ① *Modality Forgetting Issue.* Diverse modalities may prefer conflicting optimization directions, resulting in ineffective learning or knowledge forgetting; ② *Modality Fitting Issue.* Current SMoE pipelines select a fixed number of experts for all modalities, which can end up over-fitting to simpler modalities or under-fitting complex modalities; ③ *Heterogeneous Learning Pace.* The varied modality attributes, task resources, and objectives usually lead to distinct optimization difficulties and convergence. Given these issues, there is a clear need for a systematic approach to harmonizing multi-modal and multi-task objectives when using SMoE. We aim to address these pain points and propose a new Sparse MoE for Multi-Modal Multi-task learning, *a.k.a.*, SM^4, which (1) disentangles model spaces for different modalities to mitigate their optimization conflicts; (2) automatically determines the modality-specific model size to improve fitting; and (3) synchronizes the learning paces of disparate modalities and tasks based on training dynamics in SMoE like the entropy of routing decisions. Comprehensive experiments validate the effectiveness of SM^4, which outperforms previous state-of-the-art across 3 task groups and 11 different modalities with a clear performance margin (*e.g.*, $\geq 1.37\%$) and a substantial computation reduction ($46.49\% \sim 98.62\%$). Codes are in supplement.

1. Introduction

Multi-modal multi-task learning (*a.k.a.*, M^{3}TL) aims to resolve different objectives simultaneously. Each objective takes various modalities as input, which is a common scenario required in real-world applications like robotics [1] and auto-driving systems [2]. Many prior works have extended unimodal transformers [3] to handle multiple multi-modal tasks [4–8]. In their ideal setup, the information from different modalities and tasks prompts each other for better performance. However, the optimization complexity of this sophisticated system limits the development of effective solutions [9, 10]. Recently, the sparsely-gated Mixture-of-Experts (SMoE) method was identified as a powerful tool for these complex training dynamics of multi-task [11–16] or multi-modal [17–19] learning. SMoE selects a subset of experts for a specific task or modality per input sample and has led to state-of-the-art performance [16, 20].

Despite preliminary success in M^{3}TL, when we try to model multiple modalities and multiple tasks through a *single* network (*e.g.*, SMoE), several under-explored pain points persist:

Second Conference on Parsimony and Learning (CPAL 2025).

① *Modality Forgetting Issue.* Considering a model trained on multiple modalities, diverse modalities can prefer conflicting optimization directions within shared parameters. For instance, recent works have shown that there are negative cosine similarities between gradients from different modalities [21–24]. Such gradient disagreement within a network can lead to inferior learning, or, in the worst case, the multi-modal model can degenerate into a "single-modal" model that only learns the modality with dominant gradients [10]. It finally behave like some modalities are "forgotten" by the network after training. Note that, this differs from the "forgetting issue" discussed in [25–27], which focuses on knowledge forgetting when continuously training a model without previous training data. In contrast, we address forgetting between modalities during training, where all modalities are present throughout the training process. ② *Modality Fitting Issue.* The vanilla SMoE architecture activates a *fixed* number of experts to deal with each input. However, some modalities are easier to learn than others. Using too many experts for a simple modality may cause overfitting, while too few experts for complex modalities may cause underfitting [24]. As more modalities are introduced, this weakness likely grows. ③ *Heterogeneous Learning Pace.* Current SMoE solutions also have yet to adapt to different objectives between tasks. In reality, the objectives can vary substantially. Consider writing robots, for example. A writing robot must handle two tasks: *object pose prediction* and *digit number classification*. Pose prediction uses images, force sensors, proprioception sensors, and robotic control signals as observations to predict the object's position after the robot executes the control signal. Digit classification uses images and audio to output the corresponding number. Each objective differs significantly in terms of modality attributes, task resources, and task objectives, which leads to great heterogeneity in their optimization convergence [28–30].

In this paper, we upgrade the original SMoE algorithm for Multi-Modal Multi-task learning, herein termed SM^4, tackling the aforementioned barriers. Specifically, SM^4 facilitates learning from three perspectives: ① (*Model*) SM^4 customizes the SMoE layer into both the feed-forward networks (FFN) and multi-head self-attention modules (MSA) in transformers, which sufficiently disentangles network parameter space for different modalities and tasks. As shown in Figure 2, the gradient conflict is then greatly reduced. ② (*Routing*) An adaptive expert allocation mechanism is proposed to automatically determine the number of selected experts (or model capacity) for different modalities. SM^4 monitors the modality-specific training dynamics (*e.g.*, validation loss), which serve as a reliable indicator to activate more or less experts to mitigate possible under-fitting or over-fitting, respectively. Figure 2 shows an example of how SM^4 mitigates over-fitting in a simple modality. ③ (*Optimization*) For each modality in one task, SM^4 adopts adaptive learning paces based on the convergence status of modality-specific routing policies to synchronize the optimization of multiple objectives. Our contributions can be summarized as follows:

- ⋆ We propose SM^4, a framework for multi-modal multi-task learning, which contains tailored SMoE layers for replacing FFN and sparse mixture-of-attention layers as the alternative for vanilla MSA modules in transformers. This disentangles network parameters and alleviates gradient conflicts between different modalities and tasks.
- ⋆ We identify two essential factors in M^{3}TL, *i.e., modality fitting issue* and *heterogeneous learning pace*, which are unstudied by existing SMoE approaches. We then propose corresponding *adaptive expert allocation* and *adaptive learning paces*.
- ⋆ Extensive empirical investigations over 3 representative task groups and 11 diverse modalities consistently validate the effectiveness of SM^4. Our method surpasses dense models with similar computational costs, and shows substantial performance improvements; SM^4 outperforms existing M^{3}TL SOTA using only 1.38% to 53.51% of their computational cost.

2. Related Work

Multi-modal and Multi-task Learning. There has been a long history of work on multi-modal learning [31–43] and multi-task learning [11, 16, 44–49]. Recently, more deep learning models expect integrating different modal and different tasks into one unimodal network [4–8]. They aim to leverage knowledge or information from the diverse modalities or tasks to help each other. For

instance, VATT [8] uses a shared model on video, audio, and text data to perform audio-only, video-only, and image-text retrieval tasks, and HighMMT [50] explores modalities beyond the old-school studies of language, vision, and audio to other common modalities such as tabular, time-series, sensors, graphs, and set data, in a multi-task environment. However, there is no free lunch; unimodal networks introduce more conflicts and complexity during model training. Alamri et al. [21], Goyal et al. [36], Poliak et al. [51], Thomason et al. [52] show that increasing modalities is not always beneficial. Specifically, the input of different modalities at one optimization object can lead to opposite gradient updates [22, 53], a situation also noted with identical modalities under distinct learning tasks [23]. Furthermore, multi-modal networks are often prone to overfitting the easy modalities and impeding performance [24]. The various modalities, task resources, and objectives result in unique optimization challenges.

Sparse Mixture-of-Experts (SMoE). SMoE as a special instance of conditional computing networks [54–57], has gained increasing popularity in both vision [58–66] and language [67–73] domains. It contains a group of sub-models (*i.e.*, experts) and activates them in an input-dependent fashion. Pioneering investigations leverage its conditional computing nature to assign different model pieces to their most relevant task [11, 16, 49, 74–77] or modality [20, 78] in multi-task or multi-modal learning. To be specific, Ma et al. [74], Aoki et al. [75], Hazimeh et al. [76] introduce task-dependent routing policies to select important sub-models given a task and its input sample. Positive results are presented on small-scale uni-modal applications such as classification for medical signal process [75], digital number recognition (MNIST) [76], and recommendation system [74]. Mustafa et al. [20] explores the opportunity of SMoE in multi-modal contrastive learning. Fan et al. [11] and Kim et al. [77], Rajbhandari et al. [79], He et al. [80, 81] contribute to efficient SMoE frameworks from software-hardware co-design and system angles, respectively.

3. Methodolody

The overall procedures of $\texttt{SM}^4$ are described in Figure 1. Our proposal processes the multi-task multi-modal learning in a two-step framework. (1) *Unimodal Encoder.* We first process all modalities from multi-tasks into sequences; the *Unimodal Encoder* converts each modality into sequences with the same length and concatenates modalities along the sequence dimension for each task. We refer the details of *Unimodal Encoder* to Appendix A.1. (2) *SMoE & SMoA layer.* Then, these sequential tokens are fed into the transformer layers with SMoE and SMoA, followed by task-specific heads. In SMoE and SMoA, routers choose the most relevant experts and aggregate their features for different modalities. The number of selected experts is dynamically decided according to the in-time training dynamics via `AEA`, as detailed below. Each modality's learning pace in one task is adapted via the convergence status of the routing policy from the corresponding modality by `ALP`.

3.1. Sparse Mixture of Experts/Attention in $\texttt{SM}^4$

Sparse Mixture of Experts (SMoE) SMoE [82] has been proposed to enhance model capacity while maintaining low cost per inference. In this paper, we use SMoE to disentangle network parameter space for different modalities and tasks. The SMoE layer includes a router network $\mathcal{R}$ and several experts $f_1, f_2, \ldots, f_{\texttt{E}}$ ($a.k.a$ **expert group**), where $\texttt{E}$ denotes the number of experts. For each input embedding $\mathbf{x}$, $\mathcal{R}$ activates the top-k expert networks with the largest scores $\mathcal{R}(\mathbf{x})_i$, where i is the expert index. The SMoE can be formally denoted as follows:

$$\mathbf{y} = \sum_{i=1}^{k} \mathcal{R}(\mathbf{x})_i \cdot f_i(\mathbf{x}), \mathcal{R}(\mathbf{x}) = \texttt{TopK}(\texttt{softmax}(g(x)), k), \tag{1}$$

$$\texttt{TopK}(\mathbf{v}, k) = \begin{cases} \mathbf{v} & \textit{if } \mathbf{v} \textit{ is in the top } k \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

where $f_i(\mathbf{x})$ represents the feature produced by expert f_i, which is weighted by $\mathcal{R}(\mathbf{x})_i$ to form the final output $\mathbf{y}$. g is the learnable network within a router $\mathcal{R}$, and is commonly is a small FNN with

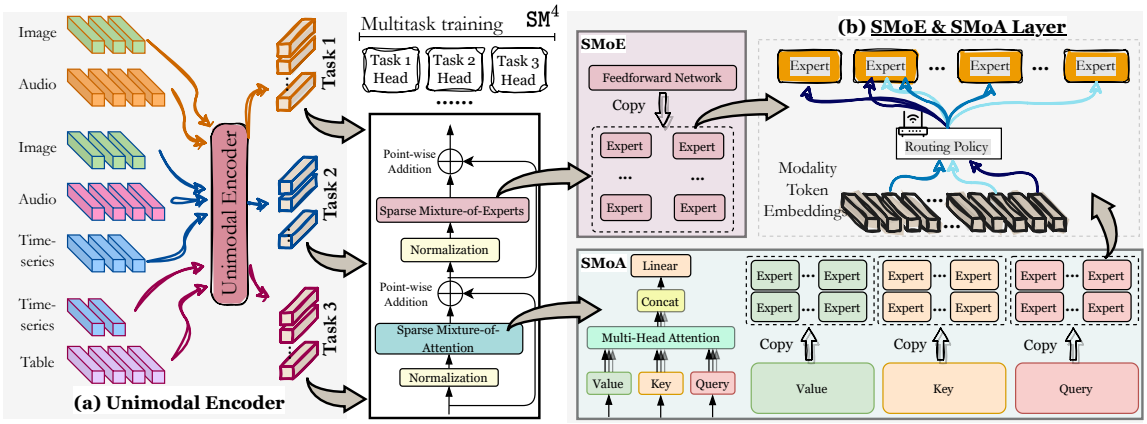

Figure 1: The overall procedure of SM^4: (a) *Unimodal Encoder.* SM^4 first standardizes each modality into a sequence, and the unimodal encoder converts each sequence to sequences of the same length. We concatenate these modality tokens on the sequence dimension within each task. Then, the transformer layers of SM^4 are performed for multi-task learning. (b) *SMoE & SMoA Layer.* Our SM^4 involves replacing FFN and MSA modules in transformers with SMoE layers and sparse mixture-of-attention (SMoA) layers that duplicate network parameters as expert group to mitigate gradient conflict.

Figure 2: Encompassing comparison between SM^4 with baseline which the dense network with the same Flops. **a**. The distribution of `cosine` distance between training gradients computed from "control" and "proprioception" modalities in PUSH dataset. The gradient is collected from the last transformer layer. More positive cosine distances denote less gradient conflict. **b**. The training loss curves, each method collects the loss curve of the "image" and "set" modalities in dataset ENRICO. **c**. The generalization gap [83] of modalities "image" and "set" in dataset ENRICO. A lower generalization gap (the difference between $\texttt{Loss}_{\texttt{training}}$, and $\texttt{Loss}_{\texttt{valid}}$) indicates better generalization performance (*i.e.*, better modality fitting). **d**. The SMoE routing distribution in dataset ENRICO. Here, we visualize the routing distribution of modality "image" in early and later iterations.

one to few [58, 82]. `TopK` sets all vector elements to zero except the elements with the largest k values. In SM^4, we duplicate the feedforward network as SMoE expert group shown in Figure 1 (b).

Sparse Mixture of Attention (SMoA) We denote the Mixture-of-Experts in the multi-head self-attention (MSA) module as Sparse Mixture-of-Attention (SMoA). As depicted in Figure 1 (b), we replicate query, key, and value layers to establish expert groups that generate query, key, and value features separately.

The SMoE and SMoA modules separate network parameter space sufficiently for different modalities and tasks. As supported by our experiments in Section 4, this model architecture mitigates gradient conflicts and enhances performance, which is more suitable for M^{3}TL. More details about SMoE and SMoA, please refer to Appendix A.7

3.2. Routing Policy Design in SM^4

Routing Policy Handling multiple modalities without conflicting gradients by disentangling parameters intrinsically relies on a successful routing policy. As shown in Figure 1, SM^4 comprises 4 expert groups within each sparse transformer block: one for SMoE and three corresponding to

the SMoE components–value, key, and query expert groups. **Each expert group shares a common policy network across all modalities and tasks.** Formally, the routing policy for the modality j is:

$$\mathcal{R}_j(\mathrm{x}) = \texttt{TopK}(\texttt{softmax}(g(x)), k_j), \tag{3}$$

where k_j is the modality-specific number of activated experts, and the network g of the router is shared across all modalities and tasks. The routing policy frequently assigns large weights to the same few experts. To combat this imbalance loading phenomenon [84], we implement the load and importance balancing loss following [82]. This effective routing policy sends modality embeddings to compatibility experts, which generate high-quality modality features. This helps to solve tasks and separate the network parameter space of different modalities and tasks. As supported by Figure 2 **a**, the disentangled model parameter space results in effective minimized gradient conflict between modalities, enjoying an improved performance (Section 4).

While SMoE and SMoA offer some benefits alone, they are not the silver bullet for multi-modal multi-task learning. Two issues still persist: ❶ *Modality Fitting Issue.* The fixed model capacity in classical SMoE design possibly leads to uneven fitting speeds across modalities. ❷ *Heterogeneous Learning Pace.* The gigantic discrepancy between tasks and modalities can lead to challenges in convergence and optimization pace. Targeting these two obstacles, we propose two interconnected solutions: First, utilizing the optimization dynamic ($e.g.$ the validation loss dynamic) to change sparse network ($e.g.$ the number of activated experts); and second, a modality-specific learning rate adaptation indicated by the sparse network training dynamic ($e.g.$ the routing entropy in SMoE or SMoA) is used improve training optimization. These two solutions link multi-modal optimization and sparse network training.

Leveraging Optimization Statistics to Guide SMoE learning: Adaptive Expert Allocation (`AEA`) The optimal fitting pace for each modality may alter significantly due to the difference in modality complexities [24]. In `SM`^{4}, modality-specific k_j the number of activated experts determines the network size of each modality. However, manually computing k_j for increasing tasks and modalities raises training costs and risk errors (inappropriate k_j can aggravate overfitting or underfitting for each modality).

Therefore, we adopt an automatic algorithm `AEA` to determine an appropriate k_j for specific modalities in a data-driven manner. As shown in Figure 2 **c**, we can tune k_j according to the modality-specific validation loss. When the validation loss stops decreasing, we increase the activated model size by increasing k. After several training iterations, if the validation loss is still larger than the previous best validation loss, we reduce the selected expert number k for the modality. Ultimately, the modality-specific k is adopted until the end of training. We show the details of `AEA` in Algorithm 2. Figure 2 **c** shows the `AEA` effectively addressing the *Modality Fitting Issue*.

Leveraging SMoE Statistics to Guide Network Training: Adaptive Learning Pace (`ALP`) The remaining convergence and optimization pace asynchronization is addressed by our proposed `ALP`. As observed in Figure 2 **d**, the modality-specific routing policy exhibits instability in early training iterations and stabilizes over time. Therefore, we utilize the sparse network, monitoring the routing distribution entropy as an indicator of routing policy status and, accordingly, decaying the learning rate where the modality-specific routing policy entropy is high. As shown in Figure 2 **b**, `ALP` lets us align different learning paces between modalities, which synchronizes the optimization of multiple objectives. See Algorithm 4 for `ALP` details.

4. Experiments

4.1. Implementation Details

Datasets and Tasks. To assess our method, we conduct experiments on MultiBench, a large-scale multi-modal multi-task benchmark containing more than 10 modalities and 20 prediction tasks across 6 research areas. As shown in Table 4.1, we follow the HighMMT choose 7 tasks in MultiBench and train 3 multi-modal multi-task models from the combinations of these tasks for the small, medium, and large settings, respectively. For more details, see Appendix C.

Setting	Dataset	Modalities	Prediction Task	Research Area	Size
Small	PUSH	image,force,proprioception,control	object pose	Robotics	37,990
	V&T	image,force,proprioception,depth	contact	Robotics	147,000
Medium	ENRICO	image,set	design interface	HCI	1,460
	PUSH	image,force,proprioception,control	object pose	Robotics	37,990
	AV-MNIST	image,audio	digit	Multimedia	70,000
Large	UR-FUNNY	text,video,audio	humor	Affective Computing	16,514
	MOSEI	text,video,audio	sentiment	Affective Computing	22,777
	MIMIC	time-series,table	ICD-9 codes	Healthcare	36,212
	AV-MNIST	image,audio	digit	Multimedia	70,000

Table 1: We follow the setting of HighMMT [50], which uses 3 multi-modal multi-task training to evaluate the performance of the SM^4. These setups include tasks with different modality inputs, predicting objectives, research areas, and dataset size.

Baselines and Configuration Details. We consider two state-of-the-art (SOTA) baselines in multi-modal multi-task learning: MultiBench [85] and HighMMT [50]. Particularly, the released code of HighMMT is implemented to achieve the desired performance with provided hyperparameters (see Appendix A). MultiBench contains 20 different models for every task; we report the performance range of these models for each adopted task. We display our model architecture overview in Figure 1. We conduct all of our experiments on the NVIDIA A30 Tensor Core GPU. Please refer to Appendix A.6 for more details on network configuration and training setup.

Evaluation Metrics. We use the standard evaluation metrics provided by MultiBench [85]. Specifically, following [86], we use metric Δ to evaluate the performance gap between our model and baseline averaged over all the tasks in each set: $\Delta = \frac{1}{T}\sum_i^T (-1)^{l_i}(M_{m,i} - M_{b,i})/M_{b,i}$, where $M_{m,i}$ and $M_{b,i}$ denote the performances of our SM^4 and baseline model, respectively; T is the number of considered tasks; and $l_i = 1$ if a higher metric value means better performance otherwise $l_i = -1$. The results of HighMMT and SM^4 are reported by the mean of three independent runs. For the min and max performances of MultiBench, we reuse the numbers directly from its publication to have a comprehensive comparison.

4.2. Performance Comparison of SM^4 with Existing Multimodel Models

We compare our model's performance with SOTA HighMMT [50] as well as 20 multi-modal models implemented in benchmark MultiBench [85]. The comparison results are collected in Table 4.2, from which we make the following observations. ① Our SM^4 demonstrates great advantages with a clear performance margin compared to all baselines. Specifically, compared to the multi-modal multi-task model HighMMT, SM^4 achieves improvements up to 12.93%, 20.19%, and 2.28% for small, medium, and large settings, respectively. These empirical results validate the effectiveness of our model to address the cross-task conflict and assign expert sub-networks to conduct each prediction task. ② SM^4 adaptively allocates adequate amounts of model parameters and fewer FLOPS to resolve the different tasks. For example, our method uses fewer parameters compared to HighMMT in the easy, small setting, e.g., $1.38\% \sim 53.51\%$ parameter saving, while we use larger parameter budgets in the challenging medium and large settings. The required FLOPS of SM^4 is always smaller than that of HighMMT. In other words, we have more efficient inference per task. ③ SM^4 delivers significant improvements and creates SOTA performances for some tasks. Notably, at the prediction task on EN-

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
SM^4 (ours)	**71.58**	**0.475**	**71.86**	**20.19**
Multi-router SM^4	71.00	0.684	71.03	7.81
R-Multi-router SM^4	64.38	0.995	71.33	-13.48
P-Modality-router SM^4	68.72	0.786	70.70	0.54
P-Task-router SM^4	68.38	0.833	70.69	-2.25

Table 3: Comparison of routing design. SM^4 makes use of the single router; Multi-router, R-Multi-router, P-Modality-router, and P-Task-router mean the adoptions of task-specific and/or modality-specific routing networks in SMoE and SMoA, respectively. Further investigations of the combinations between the multi-routing networks and the single-routing networks are in Appendix B.

Setting	Method	Dataset	Performance	Δ(%)	# Parameters (M)	FLOPS (G)
Small	MultiBench Models	PUSH ↓ V&T	0.574 ∼ 0.290 93.30 ∼ 93.60	-	1.09 ∼ 135	5.20 ∼ 25.11
	HighMMT	PUSH ↓ V&T	0.445 96.10	0.00	0.89 0.85	5.14 32.48
	SM^4	PUSH ↓ V&T	0.331 96.33	12.93	0.27 0.25	2.59 17.38
Medium	MultiBench Models	ENRICO PUSH ↓ AV-MNIST	44.40 ∼ 51.00 0.574 ∼ 0.290 68.50 ∼ 72.80	-	0.14 ∼ 525.70	0.25 ∼ 314.13
	HighMMT	ENRICO PUSH ↓ AV-MNIST	53.10 0.600 68.48	0.00	0.58 0.63 0.52	79.48 21.60 0.95
	SM^4	ENRICO PUSH ↓ AV-MNIST	**71.58** 0.475 71.86	20.19	1.23 1.25 1.23	1.10 2.33 0.41
Large	MultiBench Models	UR-FUNNY MOSEI MIMIC AV-MNIST	60.20 ∼ 66.70 76.40 ∼ 82.10 67.60 ∼ 68.90 65.10 ∼ 72.80	-	0.19 ∼ 31.50	0.15 ∼ 21.60
	HighMMT	UR-FUNNY MOSEI MIMIC AV-MNIST	62.00 78.40 65.60 70.60	0.00	0.52 0.52 0.52 0.52	1.51 1.65 0.67 0.95
	SM^4	UR-FUNNY MOSEI MIMIC AV-MNIST	64.24 79.47 67.91 71.05	2.28	0.76 0.76 0.76 0.76	0.38 0.53 0.15 0.43

Table 2: Performance comparison, parameter usage, and FLOPS of our model, HighMMT (SOTA multi-modal multi-task learning method on MultiBench benchmark), and all the 20 models implemented in MultiBench (report their performance range) in three settings.

RICO, SM^4 obtains 20.58% improvement compared with the best-performing model on MultiBench. Limited by the space, we report more detailed comparisons in Table 10.

4.3. Detailed Investigations of SM^4

Ablation Study: Single-router v.s. Multi-router. Unlike the routing policy design in SM^4, we notice that earlier works have investigated task-specific or modality-specific routing networks in learning the routing policy individually for different modalities or tasks in MTL [74–77]. Therefore, we ask ***What kind of routing policy is suitable for M^{3}TL?*** Under the medium setting with framework SM^4, we experiment with 4 multi-router designs to identify the optimal routing policy. We use modality-specific routers in SMoA and task-specific routers in SMoE, which are named as Multi-router SM^4. Alternatively, in R-Multi-router SM^4, we utilize modality-specific routers in SMoE and task-specific routers in SMoA. In SMoA and SMoE, we employed task-specific routers as P-Task-router SM^4, and modality-specific routers as P-Modality-router SM^4, respectively. From our results in Table 3, ① we observe that the adopted single router consistently outperforms the other

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
SM^4 (ours)	**71.58**	**0.475**	**71.86**	**20.19**
- w/o SMoA	69.06	1.227	70.26	−23.92
- w/o SMoE	68.84	0.818	70.94	−1.02
Dense Model	65.98	1.342	70.49	−32.14

Table 4: Ablation of SMoE and SMoA. Notably, the "Dense Model" has the same computation cost with SM^4. The results of the dense model with the same network capacity are in Appendix B.

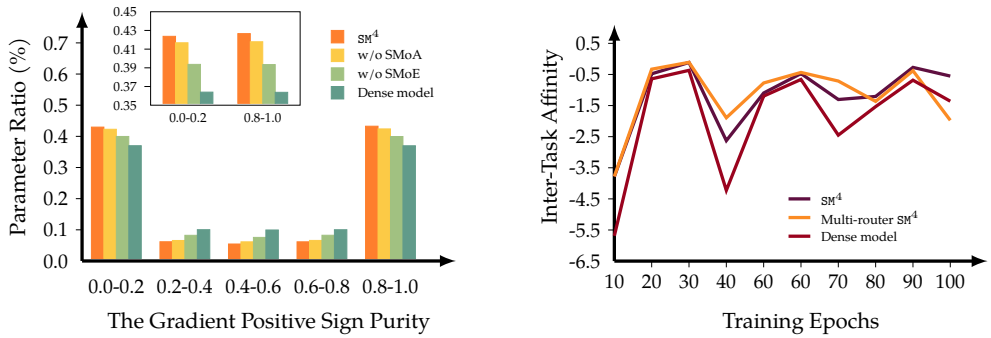

Figure 3: The distribution of Gradient Positive Sign Purity (left), and the inter-task affinity of the 'ENRICO' to the 'PUSH' task (right).

routing policies. ② Specifically, all the task-specific routers perform unpromisingly in PUSH, which contains four different data modalities. We extend this and draw similar conclusions in Appendix B.

Ablation Study: MoE. To investigate the contribution of MoE, the ablation studies are conducted with SM^4 on the medium setting. In particular, we consider three ablated models: *SM^4 w/o SMoA*: removing SMoA from the MSA module. *SM^4 w/o SMoE*: removing SMoE from FFN layer. *Dense model*: using the same computation cost with SM^4 but without any MoE components. From Table 4, we make the following observations: ① Compared with SM^4, the ablation of any MoE component significantly discounts model performance. Specifically, discarding SMoA has a more acute drop compared to discarding SMoE. This verifies our motivation for applying SMoA, which improves the model's routing capability. ② Compared with the dense model, SM^4 achieves a noticeable performance gain (*e.g.*, $\geq 1.37\%$), suggesting the benefit from the distanglement of parameter spaces. More MoE explorations can be found in Table 7.

In-Depth Discussion: Do our proposals address the gradient conflict between modalities and tasks? Yes, SM^4 is specialized to disentangle the task conflict by harmonizing the updating gradient of different tasks. We examine the following two metrics.

▷ *Gradient positive sign purity* (*GPSP*). This metric quantifies the direction consistency of backward gradients of different tasks [23]. Mathematically, we denote GPSP as $\mathcal{P}$ and record the gradient of task i as $\nabla \mathtt{W}_i$. Metric GPSP is defined as $\mathcal{P} = \sum_i \nabla \mathtt{W}_i / \sum_i |\nabla \mathtt{W}_i|$, which is further bounded into range $[0, 1]$. Specifically, $\mathcal{P}$ with a value closing to 0 or 1 indicates that the gradients from different tasks are not acutely contradictory to each other. We compare GPSP distributions of SM^4, SM^4 without MoE on self-attention, SM^4 without MoE on FFN, and the dense model. In Figure 3, we discretize the values of $\mathcal{P}$ into 5 intervals and then count the number of parameters that fall within each interval. Compared with other models, the GPSP values of SM^4 are accumulated more at the intervals of $[0.6, 0.8]$ and $[0.8, 1.0]$. This validates the effectiveness of splitting the parameter space, where only a small fraction of conflicting parameters are running for specific tasks.

▷ *Inter-task affinity*. We denote inter-task affinity with $Z_{i \to j}$, which is the influence of parameter update from task i to task j [87]. The higher value of $Z_{i \to j}$ means the parameter update is positive for task j; otherwise, task j suffers from an antagonistic updating. On the medium setting, we compare the inter-task affinity of task ENRICO to task PUSH for three backbones: SM^4, multi-router SM^4, and dense model. As shown in the right part of Figure 3, we observe the inter-task affinities of SM^4 and multi-router SM^4 tend to be higher than that of the dense model. This finding shows that MoE can restrain the gradient conflict of MTL. For more discussions on GPSP and inter-task affinity, please refer to Appendix C.4 and Appendix C.5.

In-depth Discussion: Whether our proposals address the fitting issue between modalities? Yes, we examine this question by visualizing the training loss dynamic (second subfigure) and generalization gap dynamic (third subfigure) in Figure 2. Note that the generalization gap is defined by the difference between training loss and validation loss, where a higher value means a good generalization performance on the validation set. ① *It is observed baseline HighMMT underfits in specific modality, which*

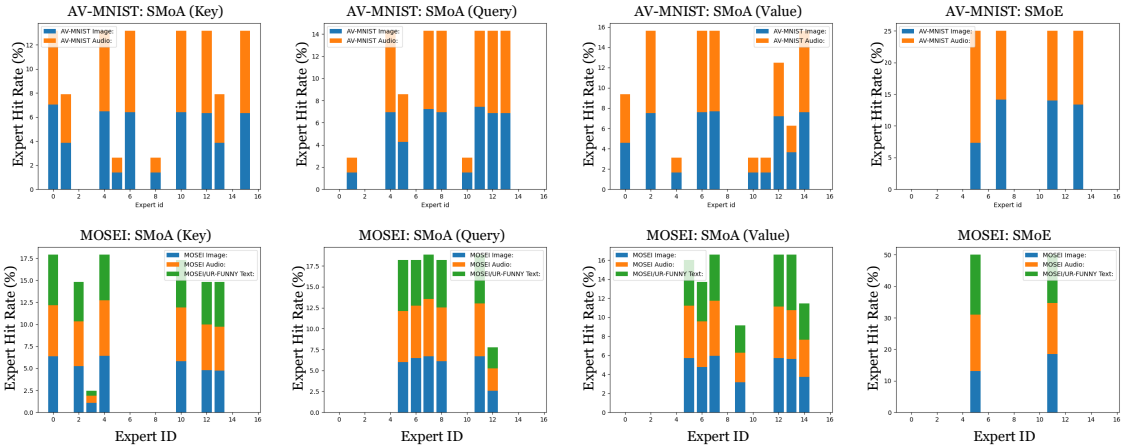

Figure 4: Analysis on the expert selection visualization produced by the SM^4 in the large setting. The first row shows the expert selection in the "AV-MNIST" dataset, and the second row shows the expert selection in the "MOSEI" dataset. Please refer to Appendix C.8 for more visualization results.

has the highest training loss accompanied with a lower generalization gap. In contrast, another modality is gradually overfitted along with the training process in HighMMT. ② *SM^4 delivers comparably superior results in all the modalities, which addresses the key challenge of under/over-fitting in multi-modal learning.* SM^4 consistently outperforms highMMT by obtaining superior generalization gaps in all modalities.

In-depth Discussion: Is the expert selection specialized to the different modalities and tasks? We show the routing distributions for different modalities and tasks of the medium setting in Figure 4, from which we make the following observations. ① *There is an overall balanced loading across the different modalities in SMoA, but it shows an imbalance in some of the experts in SMoE.* For example, expert 5 prefers modality audio in the AV-MNIST task and prefers modalities of audio as well as text in the MOSEI task. ② *The expert selection is specialized to the different tasks*. Considering the SMoA (Query) layer, we observe the AV-MNIST task leverages the unique experts 4 and 13 while the MOSEI task activates expert 6. These empirical studies show SM^4 can optimize how many (*i.e.*, adaptive network capacity) and which (*i.e.*, dynamic routing) experts to activate for each task and modality.

Additional in-depth analysis and implementation details can be found in Appendix C.

5. Conclusion and Limitation

This paper introduces SM^4, utilizing Sparse Mixture-of-Experts to address the forgetting, fitting, and learning issues in multi-modal multi-task learning. By ingeniously tailoring the Mixture-of-Experts into both the self-attention and the feed-forward networks of a transformer backbone, we achieve the following. First, the Sparse Mixture-of-Attention (SMoA) and the Sparse Mixture-of-Experts (SMoE) efficiently and sufficiently disentangle the network parameter space to mitigate the gradient conflict between different modalities and tasks. Second, we ingeniously design an adaptive expert allocation mechanism to determine the optimal number of selected experts in use for different modalities, resulting in harmonized and unified fitting speeds between modalities. Third, we adeptly adapt the learning pace by considering the convergence status of modality-specific routing policies to effectively synchronize the learning paces of different modalities and tasks. Comprehensive experiments demonstrate that the proposed SM^4 significantly surpasses the SOTA with a fraction of the computation cost ($+12.93\%/+20.19\%/+2.28\%$ M^{3}TL performance); our computation cost is only $1.38\% \sim 53.51\%$ of the SOTA model. Our experiments on MoE also provide insightful and rational perspectives for designing multi-modal multi-task learning neural network architectures. The limitation of our work is that the proposed SM^4 is only evaluated on academic datasets. Moving forward, we plan to evaluate SM^4 on more practical tasks like in-door robots and autonomous vehicles. Lastly, we anticipate expanding our model size for larger-scale tasks and incorporating more kinds of modalities in future work.

References

[1] Charles Sun, Jedrzej Orbik, Coline Manon Devin, Brian H Yang, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully autonomous real-world reinforcement learning with applications to mobile manipulation. In *Conference on Robot Learning*, pages 308–319. PMLR, 2022.

[2] Dong-Gyu Lee. Fast drivable areas estimation with multi-task learning for real-time autonomous driving assistant. *Applied Sciences*, 11(22):10713, 2021.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[4] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: pre-training of generic visual-linguistic representations. In *ICLR*. OpenReview.net, 2020.

[5] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR, 2021. URL `http://proceedings.mlr.press/v139/cho21a.html`.

[6] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *ICCV*, pages 1419–1429. IEEE, 2021.

[7] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, pages 13–23, 2019.

[8] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: transformers for multimodal self-supervised learning from raw video, audio and text. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24206–24221, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/cb3213ada48302953cb0f166464ab356-Abstract.html`.

[9] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[10] Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8238–8247, 2022.

[11] Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Advances in Neural Information Processing Systems*, 35:28441–28457, 2022.

[12] Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Ran Yi, Shouhong Ding, and Lizhuang Ma. Adaptive mixture of experts learning for generalizable face anti-spoofing. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6009–6018, 2022.

[13] Shashank Gupta, Subhabrata Mukherjee, Krishan Subudhi, Eduardo Gonzalez, Damien Jose, Ahmed H Awadallah, and Jianfeng Gao. Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*, 2022.

[14] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

[15] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018.

[16] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11828–11837, 2023.

[17] Sheng Shen, Zhewei Yao, Chunyuan Li, Trevor Darrell, Kurt Keutzer, and Yuxiong He. Scaling vision-language models with sparse mixture of experts. *arXiv preprint arXiv:2303.07226*, 2023.

[18] Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, et al. Flan-moe: Scaling instruction-finetuned language models with sparse mixture of experts. *arXiv preprint arXiv:2305.14705*, 2023.

[19] Yiming Sun, Bing Cao, Pengfei Zhu, and Qinghua Hu. Moe-fusion: Instance embedded mixture-of-experts for infrared and visible image fusion. *arXiv preprint arXiv:2302.01392*, 2023.

[20] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *CoRR*, abs/2206.02770, 2022. doi: 10.48550/arXiv.2206.02770. URL `https://doi.org/10.48550/arXiv.2206.02770`.

[21] Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, et al. Audio visual scene-aware dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7558–7567, 2019.

[22] Adrián Javaloy, Maryam Meghdadi, and Isabel Valera. Mitigating modality collapse in multimodal vaes via impartial optimization. In *International Conference on Machine Learning*, pages 9938–9964. PMLR, 2022.

[23] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020.

[24] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *CVPR*, pages 12692–12702. Computer Vision Foundation / IEEE, 2020.

[25] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the Catastrophic Forgetting in Multimodal Large Language Model Fine-Tuning. In *Conference on Parsimony and Learning*, pages 202–227. PMLR, January 2024.

[26] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2021.3057446.

[27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv:1612.00796 [cs, stat]*, January 2017.

[28] Yu Zhang and Dit-Yan Yeung. Multi-task learning in heterogeneous feature spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 574–579, 2011.

[29] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740, 2020.

[30] Dimitrios Kollias, Viktoriia Sharmanska, and Stefanos Zafeiriou. Distribution matching for heterogeneous multi-task learning: a large-scale face study. *arXiv preprint arXiv:2105.03790*, 2021.

[31] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. A new baseline for image annotation. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III 10*, pages 316–329. Springer, 2008.

[32] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. 2011.

[33] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[34] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. *Advances in neural information processing systems*, 26, 2013.

[35] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[36] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *CoRR*, abs/2205.11487, 2022.

[39] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. VQA: visual question answering. *Int. J. Comput. Vis.*, 123(1):4–31, 2017.

[40] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] Yong Dai, Duyu Tang, Liangxin Liu, Minghuan Tan, Cong Zhou, Jingquan Wang, Zhangyin Feng, Fan Zhang, Xueyu Hu, and Shuming Shi. One model, multiple modalities: A sparsely activated approach for text, sound, image, video and code. *CoRR*, abs/2205.06126, 2022. doi: 10.48550/arXiv.2205.06126. URL `https://doi.org/10.48550/arXiv.2205.06126`.

[42] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021. URL `http://proceedings.mlr.press/v139/jaegle21a.html`.

[43] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

[44] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(1), 2007.

[45] Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. In *ICCV*, pages 1375–1384. IEEE, 2019.

[46] Amir Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722. Computer Vision Foundation / IEEE Computer Society, 2018.

[47] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *ACL (2)*. The Association for Computer Linguistics, 2016.

[48] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In *EMNLP*, pages 1923–1933. Association for Computational Linguistics, 2017.

[49] Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. *CoRR*, abs/2307.15324, 2023.

[50] Paul Pu Liang, Yiwei Lyu, Xiang Fan, Shentong Mo, Dani Yogatama, Louis-Philippe Morency, and Ruslan Salakhutdinov. Highmmt: Towards modality and task generalization for high-modality representation learning. *CoRR*, abs/2203.01311, 2022.

[51] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*, 2018.

[52] Jesse Thomason, Daniel Gordon, and Yonatan Bisk. Shifting the baseline: Single modality performance on visual navigation & qa. *arXiv preprint arXiv:1811.00613*, 2018.

[53] Hassan Akbari, Dan Kondratyuk, Yin Cui, Rachel Hornung, Huisheng Wang, and Hartwig Adam. Alternating gradient descent and mixture-of-experts for integrated multimodal perception. *arXiv preprint arXiv:2305.06324*, 2023.

[54] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[55] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[56] Ke Chen, Lei Xu, and Huisheng Chi. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks*, 12(9):1229–1252, 1999.

[57] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012. doi: 10.1109/TNNLS.2012.2200299.

[58] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8583–8595, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/48237d9f2dea8c74c2a72126cf63d933-Abstract.html.

[59] Yuxuan Lou, Fuzhao Xue, Zangwei Zheng, and Yang You. Cross-token modeling with conditional computation. *arXiv preprint arXiv:2109.02008*, 2021.

[60] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

[61] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. Network of experts for large-scale image categorization. In *European Conference on Computer Vision*, pages 516–532. Springer, 2016.

[62] Sam Gross, Marc'Aurelio Ranzato, and Arthur Szlam. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6865–6873, 2017.

[63] Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR, 2020.

[64] 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.

[65] Alhabib Abbas and Yiannis Andreopoulos. Biased mixtures of experts: Enabling computer vision inference under data transfer limitations. *IEEE Transactions on Image Processing*, 29: 7656–7667, 2020.

[66] Svetlana Pavlitskaya, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and Marius Zollner. Using mixture of expert models to gain insights into semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 342–343, 2020.

[67] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=qrwe7XHTmYb`.

[68] Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andrés Felipe Cruz-Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. Scalable and efficient moe training for multitask multilingual models. *CoRR*, abs/2109.10465, 2021. URL `https://arxiv.org/abs/2109.10465`.

[69] 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.

[70] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. *arXiv preprint arXiv:2202.09368*, 2022.

[71] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*, 2021.

[72] Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. Taming sparsely activated transformer with stochastic experts. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=B72HXs80q4`.

[73] Hao Jiang, Ke Zhan, Jianwei Qu, Yongkang Wu, Zhaoye Fei, Xinyu Zhang, Lei Chen, Zhicheng Dou, Xipeng Qiu, Zikai Guo, et al. Towards more effective and economic sparsely-activated model. *arXiv preprint arXiv:2110.07431*, 2021.

[74] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1930–1939. ACM, 2018. doi: 10.1145/3219819.3220007. URL https://doi.org/10.1145/3219819.3220007.

[75] Raquel Aoki, Frederick Tung, and Gabriel L. Oliveira. Heterogeneous multi-task learning with expert diversity. *CoRR*, abs/2106.10595, 2021. URL https://arxiv.org/abs/2106.10595.

[76] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29335–29347, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f5ac21cd0ef1b88e9848571aeb53551a-Abstract.html.

[77] Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andrés Felipe Cruz-Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. Scalable and efficient moe training for multitask multilingual models. *CoRR*, abs/2109.10465, 2021. URL https://arxiv.org/abs/2109.10465.

[78] Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3577–3599. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.304. URL https://doi.org/10.18653/v1/2021.findings-emnlp.304.

[79] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In *International Conference on Machine Learning*, pages 18332–18346. PMLR, 2022.

[80] Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang. Fastmoe: A fast mixture-of-expert training system. *arXiv preprint arXiv:2103.13262*, 2021.

[81] Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. Fastermoe: modeling and optimizing training of large-scale dynamic pre-trained models. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 120–134, 2022.

[82] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

[83] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. In *ICLR (Poster)*. OpenReview.net, 2019.

[84] Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35:34600–34613, 2022.

[85] Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A. Lee, Yuke Zhu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Multibench: Multiscale benchmarks for multimodal representation learning. In *NeurIPS Datasets and Benchmarks*, 2021.

[86] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3614–3633, 2022. doi: 10.1109/TPAMI.2021.3054719.

[87] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *NeurIPS*, pages 27503–27516, 2021.

[88] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[89] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

[90] Michelle A. Lee, Brent Yi, Roberto Martín-Martín, Silvio Savarese, and Jeannette Bohg. Multimodal sensor fusion with differentiable filters. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021*, pages 10444–10451. IEEE, 2020. doi: 10.1109/IROS45743.2020.9341579. URL https://doi.org/10.1109/IROS45743.2020.9341579.

[91] Michelle A. Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Trans. Robotics*, 36(3):582–596, 2020. doi: 10.1109/TRO.2019.2959445. URL https://doi.org/10.1109/TRO.2019.2959445.

[92] Luis A. Leiva, Asutosh Hota, and Antti Oulasvirta. Enrico: A dataset for topic modeling of mobile UI designs. In Susanne Boll and Simon T. Perrault, editors, *MobileHCI '20: 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services: Expanding the Horizon of Mobile Interaction, Extented Abstracts, Oldenburg, Germany, October 5-9, 2020*, pages 9:1–9:4. ACM, 2020. doi: 10.1145/3406324.3410710. URL https://doi.org/10.1145/3406324.3410710.

[93] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. Centralnet: A multilayer approach for multimodal fusion. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, volume 11134 of *Lecture Notes in Computer Science*, pages 575–589. Springer, 2018. doi: 10.1007/978-3-030-11024-6_44. URL https://doi.org/10.1007/978-3-030-11024-6_44.

[94] Amir Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2236–2246. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1208. URL https://aclanthology.org/P18-1208/.

[95] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[96] Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. Uni-perceiver-moe: Learning sparse generalist models with conditional moes, 2022. URL https://arxiv.org/abs/2206.04674.

A. Model details

A.1. Process Data into Sequence

Following the process of [43], we first standardize each input into a sequence. For each modality [43], we define some hyperparameters (such as max_freq, num_freq_bands, and freq_base) for the Fourier positional encoding. Fourier transformations get this positional information. For modalities such as text and time-series, they are already sequential data. We apply 1D positional encoding for these modalities $x \in \mathbb{R}^{b_m \times t_m \times d_m}$, where b_m, t_m, d_m are the batch size, sequence length, and input dimension of current modality, respectively. For image and similar modalities, we follow the processing procedure of [88], which breaks each input into $h_m \times w_m$ patches and flattens it as a sequence of p^2 regions. We use 2D positional encoding for image and similar modalities input $x \in \mathbb{R}^{b_m \times h_m \times w_m \times d_m}$, where $h_m \times w_m$ is the number of patches. For image modality, the d_m is the number of pixels within a patch. For video and similar modalities, we treat each frame data as the image modality, therefore we apply 3D positional encoding for input $x \in \mathbb{R}^{b_m \times l_m \times h_m \times w_m \times d_m}$, where l_m is the number of the frame. In the other modalities, such as table and graph, we treat each element in the table/graph as an element in the sequence and use a 1D positional encoding.

After transposing inputs into sequence data, we show the subsequent processing procedure in Algorithm 1. The 'max_modality_dim' equals to $\max_{m \in M}(d_m + d_{pm})$, where d_{pm} is the dimension of Fourier positional encoding for the corresponding modality. The one-hot encoding is defined as $e_m \in \mathbb{R}^{|M|}$, where $|M|$ is the number of all modalities involved.

Algorithm 1 Data Preprocess in Python style

```
# x: the input tokens of specific modality
def data_preprocess(x,modality, max_modality_dim):
    # get positional encoding information
    # pos_dim: indicates 1D/2D/3D positional encoding
    enc_pos = fourier_encode(modality.pos_dim,
                             modality.max_freq,
                             modality.num_freq_bands,
                             modality.freq_base)
    # add padding for modalities with smaller input dimension
    # max_modality_dim: the maximum input dimension overall modalities
    # input_dim: the input dimension of the current modality
    padding=zeros(max_modality_dim-modality.input_dim)
    # modality one-hot encoding
    # modality_index: the index of current modality
    modality_encodings = one_hot(modality.modality_index)
    # construct final input
    modality_input = concatenate(x, padding, enc_pos, modality_encodings)
    return modality_input
```

A.2. The Unimodal Encoder

The result of Algorithm 1 is then fed into the unimodal encoder layer. We display the details of the unimodal encoder layer in Figure 5. The sequence length T of different modalities are different, as T can be t_m, $h_m \times w_m$, or $l_m \times h_m \times w_m$. However, the cross-attention between the input sequence and latent input will convert the sequence length from different modalities into the same value. For example, the input modality sequence is $x \in \mathbb{R}^{T_m \times D}$ and the latent input is $z \in \mathbb{R}^{N \times C}$. After these three linear layer, we got $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{T_m \times X}$ and $\mathbf{Q} \in \mathbb{R}^{N \times X}$. Following the scaled-dot product attention:

$$\textit{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \textit{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{C}})\mathbf{V}, \tag{4}$$

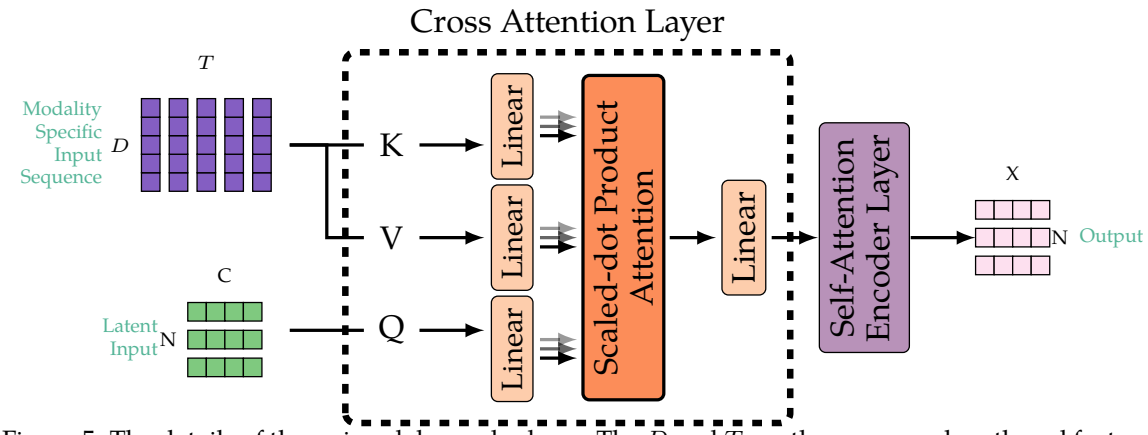

Figure 5: The details of the unimodal encoder layer. The D and T are the sequence length and feature dimension of the modality-specific input sequence. The N and C are the sequence length and the number of dimensions of latent input. The latent input is the learnable parameters shared across different modalities and tasks.

from which we can know the dimension after the attention is $Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{N \times X}$. Therefore, the sequence length of the output depends on the sequence length of the latent input, and the feature dimension depends on the unimodal encoder's hidden size, which is independent of the shape of the input modality sequence. The hidden dimension of the self-attention encoder layer equals the previous layer's cross-attention layer.

A.3. Detials of Adaptive Expert Allocation

The python style pseudo code for the Adaptive Expert Allocation (`AEA`) algorithm 2. The `AEA` is executed during the multi-modal multi-task learning.

Algorithm 2 Adaptive Expert Allocation (`AEA`) in Python style

```
def adaptive_expert_allocation(modality_set, model, modality_topk, loss):
    for modality in modality_set:
        # If the modality is signed improved, skip this modality
        if check(modality):
            continue
        n_experts = modality_topk[modality]
        loss_val = loss[valid][modality]
        # if the expert number of this modality is increased last time
        if increase_expert(modality):
            if loss_decrease(loss_val):
                improved = True
        else:
            # if the valid loss does not decrease
            if loss_decrease(loss_val):
                if not improved:
                    Sign this modality as improved
                    modality_topk[modality] = n_experts - 1
                else:
                    modality_topk[modality] = n_experts + 1
                    improved = False
    return modality_topk
```

Algorithm 3 An algorithm with caption

Require: Expert networks $f_i(i \in \{1, 2, ..., E\})$.
Require: $m \in \{1, 2, ..., M\})$ is the modality index, and M the number of modalities
Require: The validation set $\mathcal{D}$.
Require: The objective function for each modality $\mathcal{L}_{val}^{m}$ on the validation set.
Set the indicator for each modality as $Improved_m = True$;
Set the best validation loss for each modality initialize as $\mathcal{L}_{val(best)}^{m} = \infty$;
Set the number of selected experts for each modality as $k_m = 1$;
Set the modality align sign $align_m = True$;
for each epoch **do**
    1 epoch multi-modal multi-task training;
    Calculate the validation loss for each modality $\mathcal{L}_{val(best)}^{m}(m \in \{1, 2, ..., M\})$;
    for a given modality m **do**
        if $align_m$ **then**
            if $\mathcal{L}_{val(best)}^{m} \leq \mathcal{L}_{val}^{m}$ **then**
                if not $Improved_m$ **then**
                    $align_m$=False;
                    break;
                else
                    $k_m = k_m + 1$;
                    $Improved_m$=False;
                end if
            else
                $\mathcal{L}_{val(best)}^{m} = \mathcal{L}_{val}^{m}$;
                $Improved_m$=True;
            end if
            if not $align_m$ **then**
                $k_m = k_m - 1$;
            end if
        end if
    end for
end for

A.4. Details of Adaptive Learning Pace

The python style pseudo code for the Adaptive Learning Pace (`ALP`) algorithm 4. The `ALP` adjusts the learning pace of each modality by out modality-specific learning rate weights.

Algorithm 4 Adaptive Learning Pace (`ALP`) in Python style

```
def adaptive_learning_pace(routing_entropy):
    return 1 - routing_entropy.softmax()
```

A.5. Overall Progress

Algorithm 5 Overall Training in Python style

```
def overall_training(modality_set, model):
    for i in range(max_epochs):
        # training 1 epoch
        # losses include the average valid loss of all
        # modalities in this epoch
        # routing_entropy includes the average entropy of
        # all modalities in this epoch
        val_losses, routing_entropy =
                train(model, modality_topk, modality_weights)

        # Setting top-k of all modalities
        # If the monitoring of specific modality is ended,
        # AEA will skip to tuning the top-k of specific modality
        modality_topk =
                AEA(modality_set, model, modality_topk, val_losses)
        modality_weights = ALP(routing_entropy)
```

A.6. The Model and Training Setups

We list hyperparameters for the training and the model in Table 15, Table 16, and Table 17 for small, medium and large settings, respectively.

A.7. Expert Group

Our framework comprises four distinct expert groups: one group, situated within SMoE, consists of experts duplicating feedforward networks. Meanwhile, SMoA includes three expert groups, each dedicated to duplicating the query, key, and value networks, respectively. Experts are grouped by the nature of where they are duplicated from.

B. Single-router v.s. Multi-router in SM^4

SM^4 use SMoE and SMoA to disentangle the network parameter space. Moreover, several works [20, 74–77, 77, 78] investigate single-router or multi-router for multi-task learning or multi-modal learning. Therefore, we also investigate the multi-router SM^4 for M^{3}TL. With those in mind, we ask a much more significant question:

What kind of router design is appropriate for SM^4 to M^{3}TL?

For our proposed SM^4, we can use Single-router and Multi-router in both the self-attention and FFN layers, respectively. Meanwhile, the Multi-router can also be divided into the modality-specific Multi-router and the task-specific Multi-router. Therefore, we explore all possible combinations of the above settings SMoE and SMoA. Note that, without specifics, the router in SMoE and SMoA is a single router by default. Herein, the "single router" denotes one router in SMoE and three routers in SMoA ("single" refers to not using task/modality-specific in SMoE or SMoA). We list all explored network architectures in Table 5.

Table 5: All possible router design combinations for SM^4.

		SMoE			
		Modality-Specific Router	Task-Specific	Single-Router	w/o SMoE
SMoA	Modality-Specific Router	P-Modality-router SM^4	Multi-router SM^4	Modality-SMoA-Single-SMoE SM^4	Multi-router SM^4 w/o SMoE
	Task-Specific Router	R-Multi-router SM^4	P-Task-router SM^4	Task-SMoA-Single-SMoE SM^4	R-Multi-router SM^4 w/o SMoE
	Single-Router	Single-SMoA-Modality-SMoE SM^4	Single-SMoA-Task-SMoE SM^4	SM^4	SM^4 w/o SMoE
	w/o MoA	R-Multi-Router SM^4 w/o SMoA	Multi-Router SM^4 w/o SMoA	SM^4 w/o SMoA	Dense Model

Table 7: The results of different SMoE & SMoA router settings in the medium setting.

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
SM^4	**71.58**	**0.475**	**71.86**	**20.19**
Multi-router SM^4	71.00	0.684	71.03	7.81
R-Multi-router SM^4	64.38	0.995	71.33	−13.48
Dense Model 1	65.98	1.342	70.49	−32.14
Dense Model 2	62.56	1.400	71.40	−37.11
SM^4 w/o SMoE	68.84	0.818	70.94	−1.02
SM^4 w/o SMoA	69.06	1.227	70.26	−23.92
Multi-router SM^4 w/o SMoE	67.58	1.166	71.11	−21.06
Multi-router SM^4 w/o SMoA	65.41	1.402	70.08	−36.03
R-Multi-router SM^4 w/o SMoE	67.35	0.633	71.37	8.54
R-Multi-router SM^4 w/o SMoA	66.43	0.969	71.04	−10.89
Task-SMoA-Single-SMoE SM^4	63.81	0.952	71.02	−11.62
Modality-SMoA-Single-SMoE SM^4	69.52	0.777	71.47	1.94
Single-SMoA-Task-SMoE SM^4	67.24	0.764	71.03	1.00
Single-SMoA-Modality-SMoE SM^4	65.75	1.088	71.31	−17.77
P-Modality-router SM^4	68.38	0.786	70.70	0.54
P-Task-router SM^4	68.38	0.833	70.69	−2.25
Equal Capacity Model	64.61	0.878	69.80	−7.59

We run the above network architectures in the medium setting and report the results in Table 7. All results reported in Table 7 use the same hyperparameters in Table 16, except for the routing network setting. In particular, the 'Dense Model' is an equal computation dense model where we propose two kinds of equal computation dense model: 'Dense Model 1' uses the transformer encoder layer with double depth, and 'Dense Model 2' is 4x wider than the hidden dimension of the transformer encoder layer. To further illustrate our performance gains mainly come from our SM^4 design, we construct the same capacity model where we $\times 4$ the number of attention heads, $\times 8$ the dimension of each attention head, and $\times 32$ the hidden dimension of the FFN layer. .

We find out that the single-router is the best architecture for M^{3}TL. The second-best architecture uses the task-specific router in the SMoE and the dense layer in the FFN layer. Meanwhile, using the modality-specific router in the SMoA and the task-specific router in SMoE also seems like a reasonable choice.

For better understanding, we display the architecture of the **Multi-Router** SM^4 and the **R-Multi-Router** SM^4 in Figure 6 and Figure 7, respectively.

B.1. Ablation Study: Expert counts.

For the SMoE and SMoA layers, the total number of experts N is one of the most significant hyper-parameters. We show the detailed performance in Table 6 and observe consistent conclusion with [58, 89] that SM^4 still got benefits from more experts.

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
$N = 32$ (SM^4)	**71.58**	**0.475**	**71.86**	20.19
$N = 4$	67.92	1.250	71.33	−25.41
$N = 8$	67.69	0.975	70.93	−10.51
$N = 16$	69.75	0.771	70.45	1.89

Table 6: **Ablation studies**. Hyperparameter effects of the total number of experts (i.e., N) on SM^4.

B.2. Using consecutive SM^4

This section is used to illustrate how to use the consecutive SM^4 layer ($i.e.$, transformer layer with SMoE and SMoA design) as transformer backbone and provide more observation about how to use SM^4 while the network is getting deeper.

Our experimental results in Table 8 show:

- The performance may not be improved as the number of SM^4 layers increases.
- The location of SM^4 matters. Using SM^4 in shallow layers helps the most.

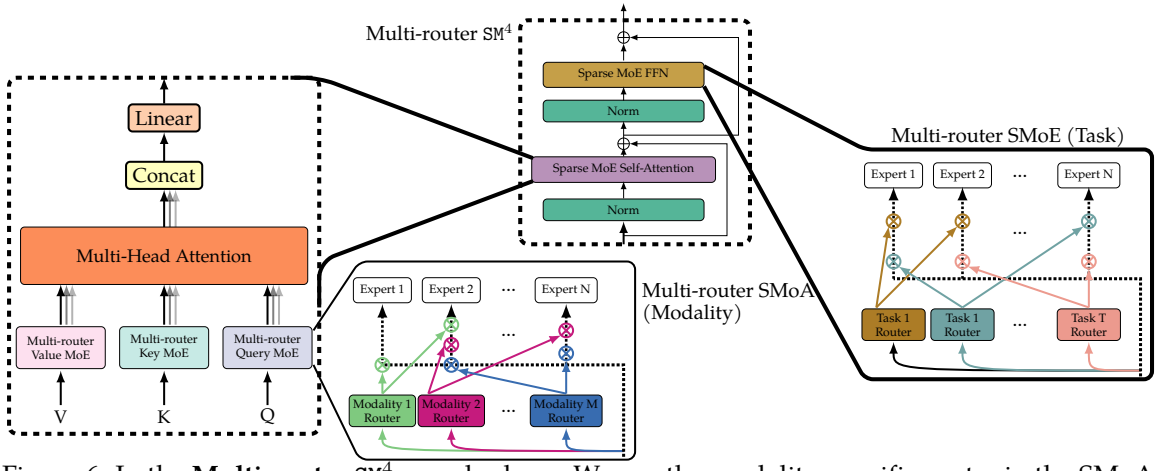

Figure 6: In the **Multi-router** SM^4 encoder layer, We use the modality-specific router in the SMoA and the task-specific router in the SMoE.

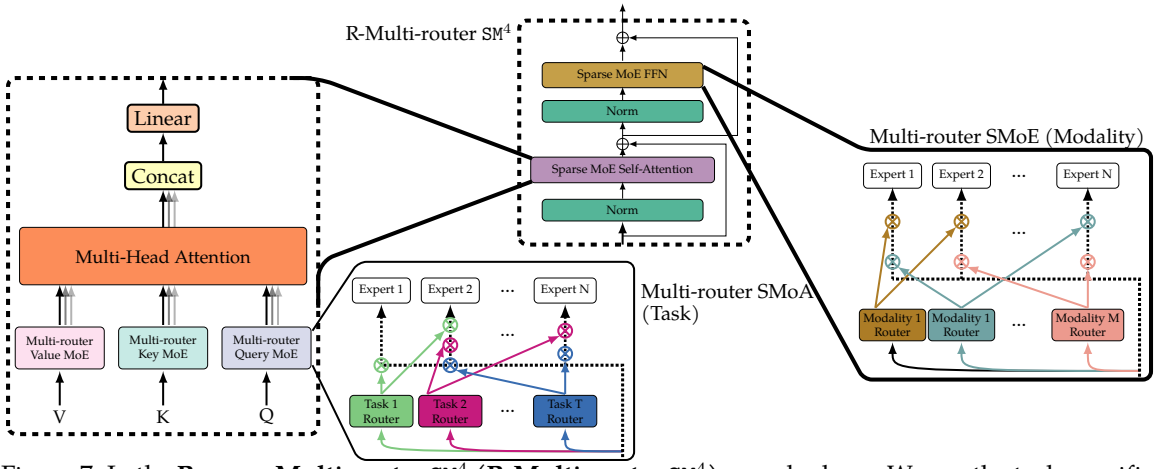

Figure 7: In the **Reverse Multi-router** SM^4 (**R-Multi-router** SM^4) encoder layer, We use the task-specific router in the SMoA and the modality-specific router in the SMoE.

C. Experiments Details

We show the number of parameters and the computation cost of the current SOTA and SM^4 in Figure 9. The "small", "medium", and "large" setting denote the number of tasks.

In Table 10, we display more comprehensive performance about MultiBench to help readers locate the position of SM^4 in the MultiBench benchmark. For each dataset, we choose multi-modal models with the best/worst performance and multi-modal models with the largest/smallest parameter numbers, respectively.

Performance explanation for HighMMT. We using the official implementation from HighMMT's repository for reporting of HighMMT performance. We strictly follow the default configurations reported in their paper, as shown in Tables 8, 9, and 10. For example, we use learning rates of 0.0005, 0.001, and 0.0008 for the small, medium, and large settings, respectively. Additionally, supplementary materials have included the HighMMT code and reproduction scripts, maintaining exact replication of their hyper-parameter settings.

Table 8: Task performances of different models. $\texttt{SM}^4$ $2/3/4$ layers: $2/3/4$ transformer encoder layers and replacing with $\texttt{SM}^4$ layer every other layer. P-$\texttt{SM}^4$ $2/3/4$ layers: $2/3/4$ consecutive $\texttt{SM}^4$ layers. $\texttt{SM}^4$ early/middle/late-2: 4 transformer encoder layers and replacing the early/middle/late-2 encoder layers with two $\texttt{SM}^4$ layers.

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
$\texttt{SM}^4$	**71.58**	**0.475**	**71.86**	**20.19**
$\texttt{SM}^4$ 2 layers	70.55	0.992	70.34	−9.92
$\texttt{SM}^4$ 3 layers	69.18	0.551	70.32	13.71
$\texttt{SM}^4$ 4 layers	71.46	1.223	70.18	−22.24
P-$\texttt{SM}^4$ 2 layers	69.63	0.766	71.57	2.64
P-$\texttt{SM}^4$ 3 layers	70.78	0.616	71.12	11.49
P-$\texttt{SM}^4$ 4 layers	67.47	0.976	71.68	−10.30
$\texttt{SM}^4$ early two layer	68.15	0.793	71.19	−0.03
$\texttt{SM}^4$ middle two layer	73.17	0.884	69.89	−2.49
$\texttt{SM}^4$ late two layer	72.15	1.374	69.97	−30.33

Table 9: Detailed results of parameter and computation cost.

Small setting	PUSH		V&T	
	Params (M)	Flops (G)	Params (M)	Flops (G)
HighMMT multitask	0.89	5.14	0.85	32.48
$\texttt{SM}^4$	0.27	2.59	0.25	17.38

Medium setting	ENRICO		PUSH		AV-MNIST	
	Params (M)	Flops (G)	Params (M)	Flops (G)	Params (M)	Flops (G)
HighMMT multitask	0.58	79.48	0.63	21.60	0.52	0.95
$\texttt{SM}^4$	1.23	1.10	1.25	2.33	1.23	0.41

Large setting	UR-FUNNY		MOSEI		MIMIC		AV-MNIST	
	Params (M)	Flops (G)	Params (M)	Flops (G)	Params (M)	Flops (G)	Params (M)	Flops (G)
HighMMT multitask	0.52	1.51	0.52	1.65	0.52	0.67	0.52	0.95
$\texttt{SM}^4$	0.76	0.38	0.76	0.53	0.76	0.15	0.76	0.43

C.1. Dataset

PUSH [90], i.e., the **MUJOCO PUSH** task, is a planar pushing task, in which a 7-DoF Panda Franka robot is pushing a circular puck with its end-effector in simulation. We estimate the 2D position of the unknown object on a table surface while the robot intermittently interacts with the object. This dataset contains 1000 training data, 10 validation data, and 100 testing data, where each data point is split into 29 sequences, and each sequence includes 16 consecutive steps.

V&T [91], also called 'VISION&TOUCH', is a real-world robot manipulation dataset that collects visual, force, and robot proprioception data for a peg insertion task. The robot is used to insert the peg into the hole. In this paper, we use this dataset to predict the manipulator whether contacts with the peg in the next step, which is a binary classification task. We follow the setting of MultiBench and use $117,600$ data points for training and the remaining $29,400$ data points for validation and testing.

ENRICO [92] includes 20 Android app design categories. Each data point consists of the app screenshot and the view hierarchy. The view hierarchy describes the spatial and structural layout of UI elements of the corresponding screenshot. During training, the view hierarchy is rendered as "wireframe", which can be viewed as a form of set data. ENRICO contains 947 data points for training, 219 data points for validation, and 292 data points for testing.

AV-MNIST [93] is a multimedia dataset that uses audio and image information to predict the digit into one of 10 classes (0-9). This dataset comprises $55,000$ training data points, $5,000$ validation data points, and $10,000$ testing data points.

Table 10: Detailed performance comparison, parameter usage, and FLOPS of our model, HighMMT, SOTA multi-modal multi-task learning method on MultiBench benchmark in three settings. For the "FLOPS(G)", "-" indicates the MultiBench does not provide official implementation. Notably, the empty FLOPS of the "MultiBench Model (MFAS)" is due to the FLOPS of "MFAS" being dynamic during training.

Setting	Method	Dataset	Performance	Δ(%)	# Parameters (M)	FLOPS (G)
Single Task	MultiBench Models (TF-LSTM)	PUSH ↓	0.574	–	23.5	25.11
	MultiBench Models (LF-LSTM)	PUSH ↓	0.290	–	1.90	14.07
	MultiBench Models (MULT)	PUSH ↓	0.402	–	14.6	19.20
	MultiBench Models (LRTF)	V&T	93.3	–	1.09	5.20
	MultiBench Models (LF)	V&T	93.6	–	1.20	5.20
	MultiBench Models (RefNet)	V&T	93.5	–	135	–
	MultiBench Models (TF)	ENRICO	46.6	–	19.3	314.13
	MultiBench Models (GradBlend)	ENRICO	51.0	–	19.3	314.13
	MultiBench Models (RefNet)	ENRICO	44.4	–	25.7	2.67
	MultiBench Models (GradBlend)	AV-MNIST	68.5	–	0.29	0.50
	MultiBench Models (MFAS)	AV-MNIST	72.8	–	0.14	–
	MultiBench Models (RefNet)	AV-MNIST	70.9	–	14.1	0.25
	MultiBench Models (EF-GRU)	UR-FUNNY	60.2	–	3.58	3.13
	MultiBench Models (MULT)	UR-FUNNY	66.7	–	2.38	3.37
	MultiBench Models (MCTN)	UR-FUNNY	63.2	–	0.19	0.17
	MultiBench Models (TF)	UR-FUNNY	61.2	–	12.2	2.67
	MultiBench Models (MCTN)	MOSEI	76.4	–	0.19	0.15
	MultiBench Models (MULT)	MOSEI	82.1	–	4.75	3.35
	MultiBench Models (LF-Transformer)	MOSEI	80.6	–	31.5	21.60
	MultiBench Models (MI-Matrix)	MIMIC	67.9	–	0.801	0.005
	MultiBench Models (LF)	MIMIC	68.9	–	0.034	0.005
	MultiBench Models (LRTF)	MIMIC	68.5	–	0.008	0.005
Small	HighMMT	PUSH ↓	0.445	0.00	0.89	5.14
		V&T	96.10		0.85	32.48
	SM^4	PUSH ↓	0.331	12.93	0.27	2.59
		V&T	96.33		0.25	17.38
Medium	HighMMT	ENRICO	53.10	0.00	0.58	79.48
		PUSH ↓	0.600		0.63	21.60
		AV-MNIST	68.48		0.52	0.95
	SM^4	ENRICO	**71.58**	20.19	1.23	1.10
		PUSH ↓	0.475		1.25	2.33
		AV-MNIST	71.86		1.23	0.41
Large	HighMMT	UR-FUNNY	62.00	0.00	0.52	1.51
		MOSEI	78.40		0.52	1.65
		MIMIC	65.60		0.52	0.67
		AV-MNIST	70.60		0.52	0.95
	SM^4	UR-FUNNY	64.24	2.28	0.76	0.38
		MOSEI	79.47		0.76	0.53
		MIMIC	67.91		0.76	0.15
		AV-MNIST	71.05		0.76	0.43

UR-FUNNY is the multi-modal affective computing dataset of humor detection in human speech. Each data point of UR-FUNNY is a video with text, visual, and acoustic modalities. We train this dataset to predict whether the current data point makes people feel positive or negative. There are $1,166$, 300, and 400 videos in the train, valid, and test data, respectively.

MOSEI [94] is the largest dataset of sentence-level sentiment analysis and emotion recognition in real-world online videos. Each video is annotated for 9 discrete emotions (angry, excited, fearful, sad, surprised, frustrated, happy, disappointed, and neutral) and a continuous emotion value (valence, arousal, and dominance). We follow the MultiBench, training this dataset as a binary classification task. We use $16,265$, $1,869$, and $4,643$ train, valid, and test data points, respectively.

MIMIC [95], i.e., the Medical Information Mart for Intensive Care III, is a freely accessible critical care database, which records ICU patient data, including time-series and other demographic variables in the form of tabular numerical data. We use this dataset for binary classification on whether the

Table 11: Concatenate tokens along the batch axis.

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
SM^4	**71.58**	**0.475**	**71.86**	**20.19**
Concate along batch	64.38	1.174	71.05	−23.57

Table 12: Using a single router to routing tokens for q, k, and v simultaneously.

Model	ENRICO ↑	PUSH ↓	AV-MNIST ↑	Δ(%) ↑
HighMMT multitask	53.10	0.600	68.48	0.00
SM^4	**71.58**	**0.475**	**71.86**	**20.19**
qkv share routers	73.51	0.936	69.28	−5.45

patient fits any ICD-9 code in group 7 (460-519). The dataset is randomly split into $28,970$, $3,621$, and $3,621$ data points for training, validation, and testing.

For more details of the above datasets, please refer to the [85] and their released website:

https://github.com/pliang279/MultiBench.

Results of HighMMT is running by [50] released code:

https://github.com/pliang279/HighMMT.

C.2. Fusion by Concatenate Tokens on The Sequence Dimension

Before we input tokens into our transformer backbone (several consecutive transformer encoder layers), we concatenate tokens on the sequence dimension. Therefore, we can fuse different modalities by the attention layer within each transformer encoder layer. To further illustrate that such an operation is necessary, we additionally train the same model but concatenate tokens along the batch axis. Our following table shows fuse modalities by concatenating tokens along the sequence axis is positive for our tasks.

Our results in Table 11 show fuse modalities by concatenating tokens along the sequence axis is positive for our tasks.

C.3. Independent Routing Policy between q, k, and v

Prior works [89, 96] also apply MoE in the attention layer. However, they all use a single router to route tokens for q, k, and v simultaneously. We think such a design lacks flexibility. Therefore, in our MoE attention layer, the router for q, k, and v is separate, which could provide a more flexible attention mechanism. In order to support the above statement, we conduct additional experiments in Table 12 to study the advantage of SM^4 v.s. Prior MoE attention design style (q, k, v using the same router in the MoE attention).

C.4. The Gradient Positive Sign Purity of SM^4

The Gradient Positive Sign Purity [23] $\mathcal{P}$ of a single parameter for T tasks is defined as:

$$\mathcal{P} = \frac{1}{2}(1 + \frac{\sum_i^T \Delta L_i}{\sum_i^T |\Delta L_i|}), \tag{5}$$

where ΔL_i is the gradient for the task i. The Gradient Positive Sign Purity is bounded by $[0, 1]$, which $\mathcal{P}$ close to 1 or 0 indicates such parameters suffer less gradient confliction from multi-task training. We use the trained model to collect the Gradient Positive Sign Purity of such a model. Then, we

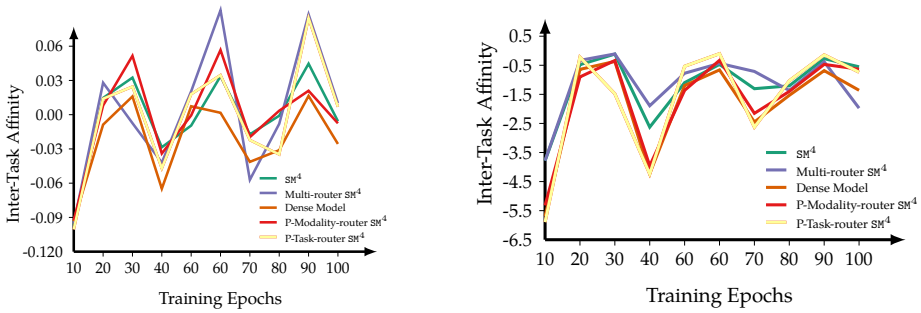

Figure 8: The inter-task affinity of the 'ENRICO' to the 'AV-MNIST' task $(left)$, and the inter-task affinity of the 'ENRICO' to the 'PUSH' task $(right)$. The results reported are the average of three replicates.

discrete the Gradient Positive Sign Purity value into five intervals of each parameter and count the ratio of parameters in these five intervals.

C.5. The Task Affinity of SM^4

The task affinity [87] is defined as follows:

$$\mathcal{Z}^t_{i\to j} = 1 - \frac{L_j(\mathcal{X}^t, \theta^{t+1}_{s|i}, \theta^t_j)}{L_j(\mathcal{X}^t, \theta^t_s, \theta^t_j)}, \tag{6}$$

where $\mathcal{X}^t$ is the training batch at time-step t, $\theta^{t+1}_{s|i}$ is the updated shared parameters after a gradient step with respect to the task i. θ^t_j represents the task j's specific parameters. For the medium setting, we collect the task affinity by solitary training the 'PUSH' task for a single epoch, and then we calculate the loss of 'ENRICO' and 'AV-MNIST' on the corresponding training data. We count the task affinity from 'PUSH' to 'ENRICO' and 'AV-MNIST' every 10 epoch during training. We display the task affinity changes with training epochs in Figure 8. The task affinity of SM^4 and multi-router SM^4 is usually higher than the one of the dense model, which indicates that the MoE we proposed alleviates the training conflict of M^{3}TL.

C.6. The Optimal Gradient Blend of SM^4

The optimal gradient blend [24] is used to re-weight the feature of each modality during multi-modal training. The optimal gradient blend will give this modality a small weight for the modality that is easy to prone to overfitting. The weight of each modality is bounded by $[0, 1]$ within a task, and the sum of all modalities for this task is 1. Therefore, the gap between different modalities within a task indicates that the modality with a smaller weight (optimal gradient blend) tends to overfit. We collect the optimal gradient blend of the corresponding trained model to determine whether our proposed model can restrain the easy model from overfitting. We use a modified version of the optimal gradient blend where the unnormalized optimal gradient blend of modality m is defined as:

$$w^{m,n}_{unnorm} = \frac{L^m_{valid}}{L^m_{valid} - L^m_{train}}, \tag{7}$$

where L^m_{valid} is the validation loss after training n epochs only using modality m, and L^m_{train} is the training loss after training n epochs only using modality m. For task i, the final optimal gradient blend we reported is:

$$w_{i,m} = \frac{w^{m,n}_{unnorm}}{\sum_m^M w^{m,n}_{unnorm}}, \tag{8}$$

where M is the number of modalities of the task i.

For M^{3}TL, the appropriate combination between modality-specific routers and task-specific routers (multi-router SM^4) helps each other better than purely using one of them (In Figure 8 and Table 13,

Table 13: The optimal gradient blend for each task under different model architectures.

Model	ENRICO		PUSH				AV-MNIST	
	image	set	image	force	proprioception	control	image	audio
SM^4	0.48	0.52	0.00	0.37	0.32	0.31	1.00	0.00
- Dense Model (w/o MoE)	0.61	0.39	0.00	0.36	0.32	0.31	1.00	0.00
- w/o Self-attention MoE	0.63	0.37	0.00	0.37	0.32	0.30	1.00	0.00
- w/o FFN MoE	0.75	0.25	0.00	0.37	0.32	0.32	1.00	0.00
multi-router SM^4	0.71	0.29	0.00	0.35	0.32	0.32	1.00	0.00
P-Modality-router SM^4	0.73	0.27	0.00	0.37	0.31	0.32	1.00	0.00
P-Task-router SM^4	0.80	0.20	0.00	0.36	0.32	0.31	1.00	0.00

Table 14: The robustness comparison between SM^4 and highMMT. We show the δ value, which is defined as the value of performance drop when missing one modality. A smaller δ value indicates better robustness against the modality missing.

UR-FUNNY	Missing Text	Missing Video	Missing Audio
SM^4	**1.16**	**4.77**	**0.92**
HighMMT	8.22	10.30	6.62

MOSEI	Missing Image	Missing Audio	Missing Text
SM^4	**0.39**	**0.82**	**0.92**
HighMMT	10.45	17.38	12.23

MIMIC	Missing Table	Missing Timeseries
SM^4	**8.71**	17.99
HighMMT	9.72	**11.67**

AV-MNIST	Missing Image	Missing Audio
SM^4	60.6	**13.60**
HighMMT	**55.86**	36.27

the Inter-Task Affinity and the optimal gradient blend of multi-router SM^4 is better than models which only use modality-specific routers (P-Modality-router SM^4) or task-specific routers (P-Task-router SM^4)).

C.7. Modality Missing Robustness

We conduct experiments to examine the robustness of SM^4 to missing modalities in the large setting. As shown in Table 14, we assess SM^4 and HighMMT with the model UR-FUNNY under three missing scenarios of missing text, video, and audio, respectively. Our results imply that SM^4 has relatively better robustness towards missing modalities.

C.8. Expert Selection Visualization

This section explores how tokens are distributed across different tasks and modalities by the routing policy of the SM^4. We show the expert selection of each routing policy under the testing distribution in Figure 9, Figure 10, and Figure 11. In these three settings, our routers work well, and most experts handle all modalities and tasks. Meanwhile, several experts focus on specific tasks.

For the large setting, we find out that the routing policy tends to route tokens to several specific experts, which also successfully proves MTL's MoE separate gradient conflict parameters. Especially for the 'MIMIC' dataset, only 2 to 4 experts are activated for this task.

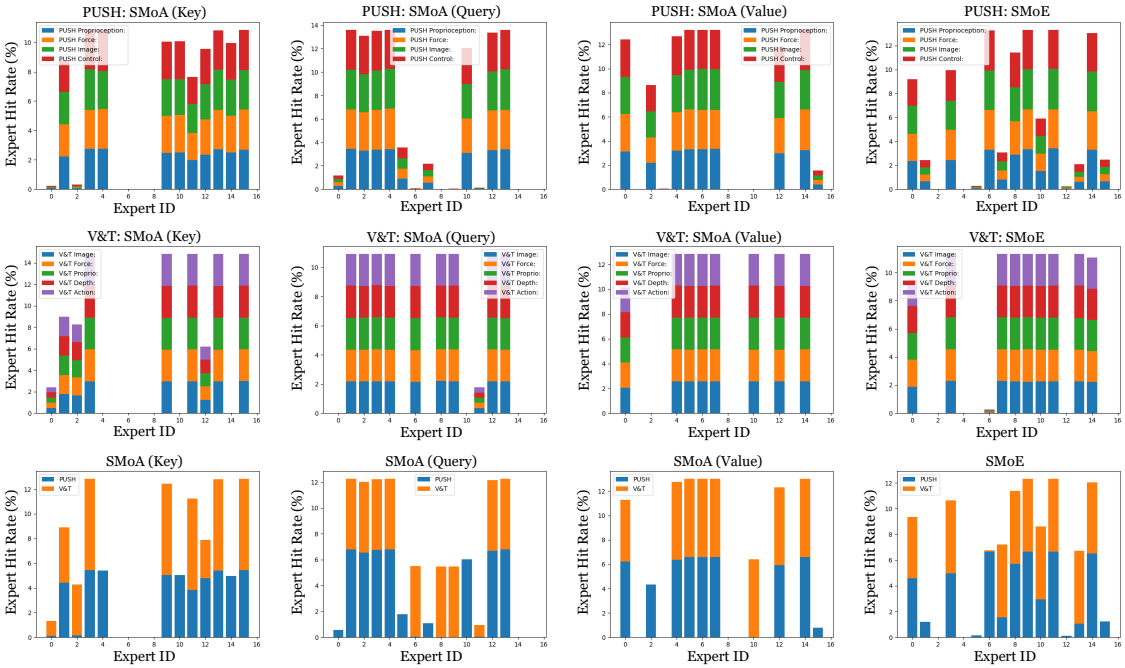

Figure 9: The expert selection of the small setting of the last $\texttt{SM}^4$ layer. The first two rows show the token distribution of different modalities for the 'PUSH' dataset and the 'V&T' dataset. The last row shows the token distribution across different tasks within three types of SMoA and SMoE.

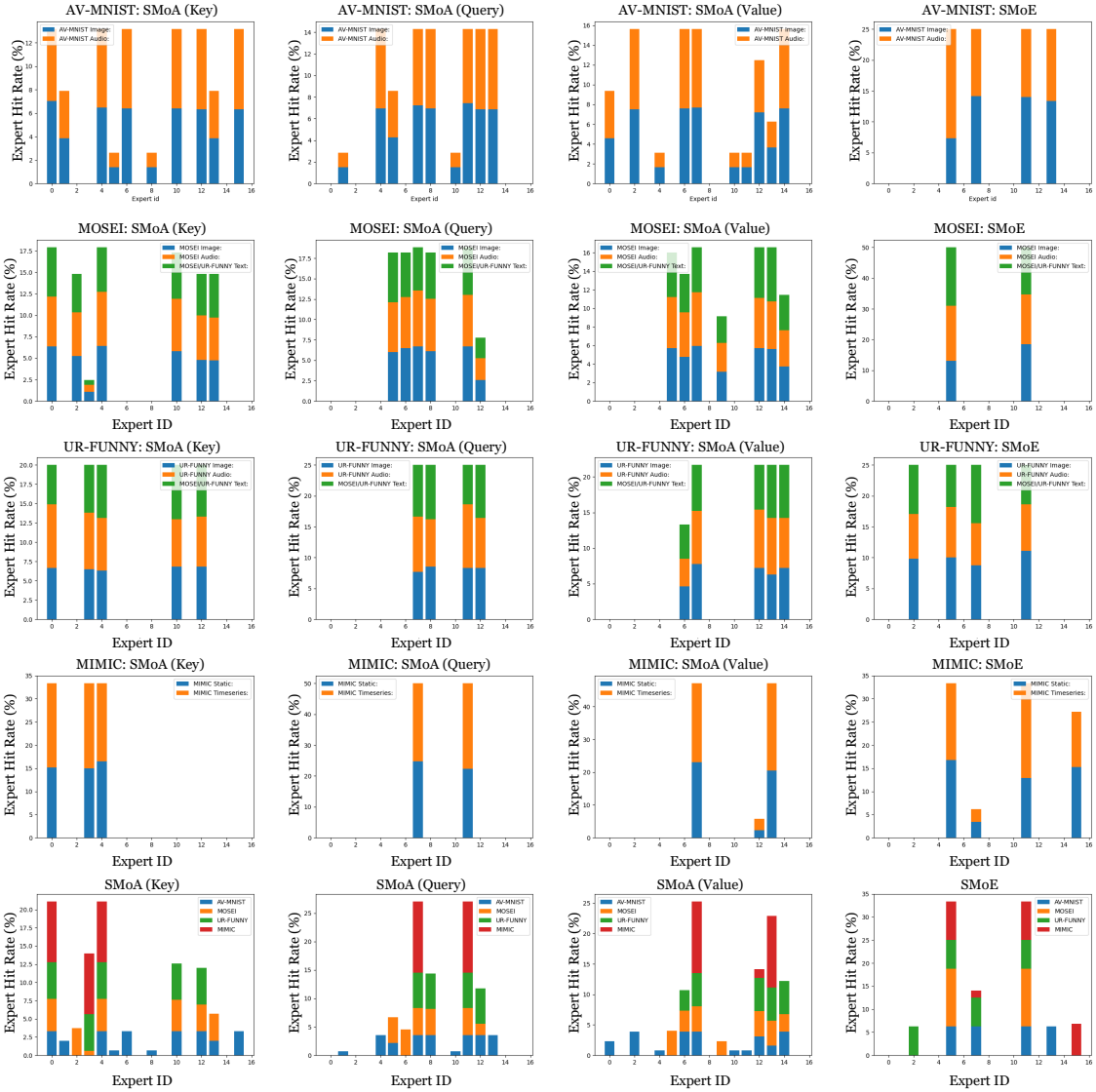

Figure 10: The expert selection of the medium setting of the last $\texttt{SM}^4$ layer. The first three rows show the token distribution of different modalities for the 'ENRICO' dataset, the 'AV-MNIST' dataset, and the 'PUSH' dataset. The last row shows the token distribution across different tasks within three types of SMoA and SMoE.

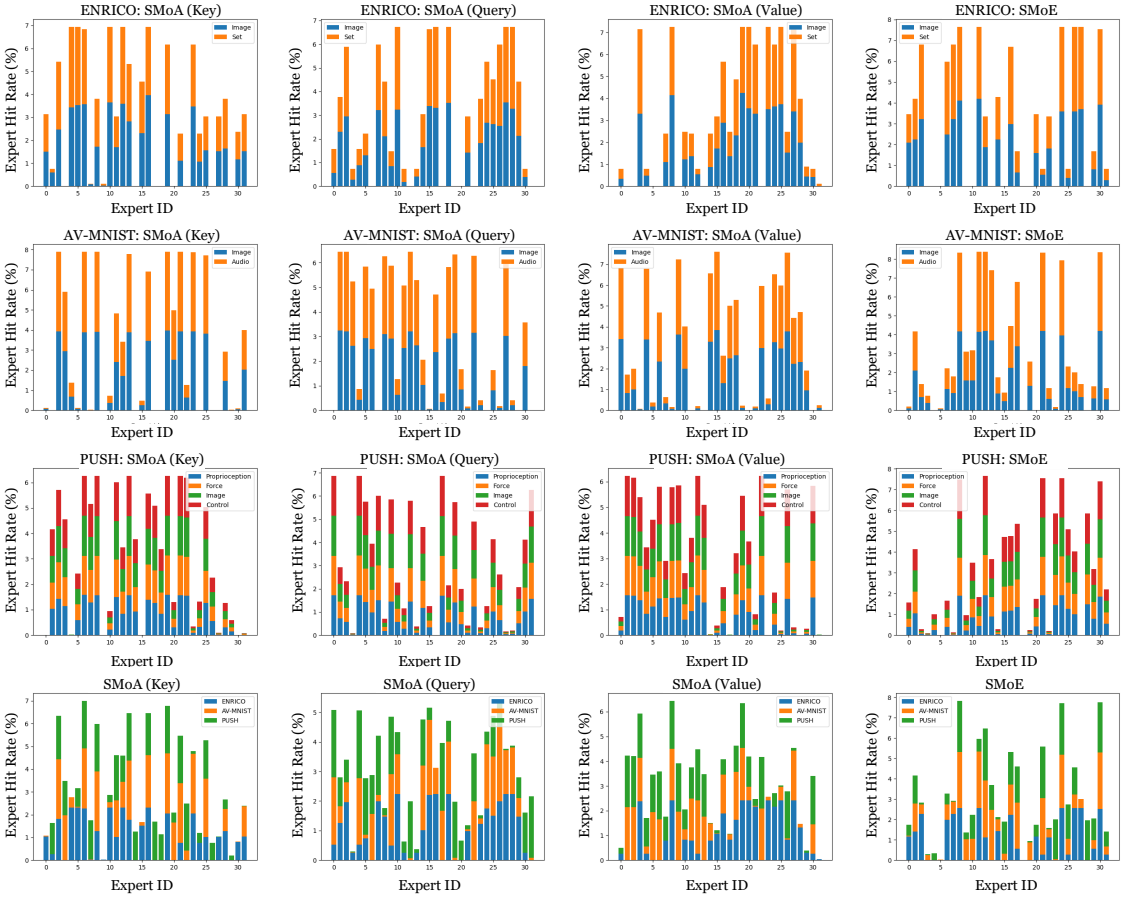

Figure 11: The expert selection of the large setting of the last $\texttt{SM}^4$ layer. The first four rows show the token distribution of different modalities for the 'AV-MNIST' dataset, the 'MOSEI' dataset, the 'UR-FUNNY' dataset, and the 'MIMIC' dataset. The last row shows the token distribution across different tasks within three types of SMoA and SMoE.

Table 15: Table of the modal and training setups on the small setting tasks: PUSH and V&T.

Model Setup			
	Name of Hyperparameter	Value	
		PUSH	V&T
Perceiver Unimodal Encoder	Sequence Length of Latent	20	
	Latent Dimension	64	
	Cross Attention Head	1	
	Cross Head Dim	64	
	Self-Attention Head	8	
	Self Head Dim	64	
MoE&MoA&Dense Encoder Layer	Depth	1	
	Self-Attention Head	8	
	Self Head Dim	8	
	Experts Number	16	
Classification Heads BatchNorm follow a Linear layer	Input/Output dimensions	256/32	320/1
Training	Optimizer	Adam	
	Learning rate	0.0005	
	Learning Scheduler	N/A	
	Weight Decay	0.0	
	Load&Importance Balancing Loss Weight	0.1	
	Pretrain	N/A	
	Max Epoch	100	
	Training loss weight	100.0	1.0
	Evaluation weight	100.0	1.0
	Batchsize	28	64
	Loss Function	MSE	CrossEntropy
MultiBench Input Dimension		Gripper Pos: 16×3 Gripper Sensors: 16×7 Image: $16 \times 32 \times 32$ Control: 16×7	Image: $128 \times 128 \times 3$ Force: 6×32 Proprio: 8 Depth: 128×128 Action: 4
Dataset	Perceiver Input Channel Size	Gripper Pos: 3 Gripper Sensors: 7 Image: 1 Control: 7	Image: 3 Force: 32 Proprio: 8 Depth: 1 Action: 4
	Perceiver Input Extra Axis	Gripper Pos: 1 Gripper Sensors: 1 Image: 3 Control: 1	Image: 2 Force: 1 Proprio: 1 Depth: 2 Action: 1
	Perceiver Input num_freq_bands	Gripper Pos: 6 Gripper Sensors: 6 Image: 6 Control: 6	Image: 6 Force: 6 Proprio: 6 Depth: 6 Action: 6
	Perceiver Input max_freq	Gripper Pos: 1 Gripper Sensors: 1 Image: 1 Control: 16×7	Image: 1 Force: 1 Proprio: 1 Depth: 1 Action: 1

Table 16: Table of the modal and training setups on the medium setting tasks: ENRICO, PUSH and AV-MNIST.

Model Setup				
	Name of Hyperparameter	Value		
		ENRICO	PUSH	AV-MNIST
Perceiver Unimodal Encoder	Sequence Length of Latent	12		
	Latent Dimension	64		
	Cross Attention Head	1		
	Cross Head Dim	64		
	Self-Attention Head	8		
	Self Head Dim	64		
MoE&MoA&Dense Encoder Layer	Depth	1		
	Self-Attention Head	8		
	Self Head Dim	8		
	Experts Number	32		
Classification Heads BatchNorm follow a Linear layer	Input/Output dimensions	128/20	256/32	128/10
Training	Optimizer	Adam		
	Learning rate	0.001		
	Learning Scheduler	CosineAnnealingLR		
	Weight Decay	0.0		
	Load&Importance Balancing Loss Weight	0.05		
	Pretrain	Training PUSH for 100 epochs first		
	Max Epoch	100		
	Training loss weight	10.0	10.0	0.8
	Evaluation weight	1.0	10.0	1.0
	Batchsize	32	32	32
	Loss Function	CrossEntropy	MSE	CrossEntropy
MultiBench Input Dimension		Image: $256 \times 128 \times 3$ Set: $256 \times 128 \times 3$	Gripper Pos: 16×3 Gripper Sensors: 16×7 Image: $16 \times 32 \times 32$ Control: 16×7	Colorless Image: 28×28 Audio Spectogram: 112×112
Dataset	Perceiver Input Channel Size	Image: 384 (cut into 16×8 rectangles) Set: 384 (cut into 16×8 rectangles)	Gripper Pos: 3 Gripper Sensors: 7 Image: 16 (cut into 4×4 squares) Control: 7	Colorless Image: 16 (cut into 4×4 squares) Audio Spectogram: 256 (cut into 16×16 squares)
	Perceiver Input Extra Axis	Image: 2 Set: 2	Gripper Pos: 1 Gripper Sensors: 1 Image: 2 Control: 1	Colorless Image: 2 Audio Spectogram: 2
	Perceiver Input num_freq_bands	Image: 6 Set: 6	Gripper Pos 6: Gripper Sensors: 6 Image: 6 Control: 6	Colorless Image: 6 Audio Spectogram: 6
	Perceiver Input max_freq	Image: 1 Set: 1	Gripper Pos: 1 Gripper Sensors: 1 Image: 1 Control: 1	Colorless Image: 1 Audio Spectogram: 1

Table 17: Table of the modal and training setups on the large setting include tasks: UR-FUNNY, MOSEI, MIMIC, and AV-MNIST.

Model Setup					
	Name of Hyperparameter	Value			
		UR-FUNNY	MOSEI	MIMIC	AV-MNIST
Perceiver Unimodal Encoder	Sequence Length of Latent	12			
	Latent Dimension	64			
	Cross Attention Head	1			
	Cross Head Dim	64			
	Self-Attention Head	8			
	Self Head Dim	64			
MoE&MoA&Dense Encoder Layer	Depth	1			
	Self-Attention Head	8			
	Self Head Dim	8			
	Experts Number	16			
Classification Heads BatchNorm follow a Linear layer	Input/Output dimensions	192/2	192/2	128/2	128/10
Training	Optimizer	Adam			
	Learning rate	0.0008			
	Learning Scheduler	N/A			
	Weight Decay	0.001			
	Load&Importance Balancing Loss Weight	0.1			
	Pretrain	N/A			
	Max Epoch	100			
	Training loss weight	0.2	1.0	1.2	0.9
	Evaluation weight	1.0	1.0	1.0	1.0
	Batchsize	32	32	20	40
	Loss Function	CrossEntropy	CrossEntropy	CrossEntropy	CrossEntropy
MultiBench Input Dimension		Image: 20×371 Audio: 20×81 Text: 50×300	Image: 50×35 Audio: 50×74 Text: 50×300	Static: 5 Time-series: 24×12	Colorless Image: 28×28 Audio Spectogram: 112×112
Dataset	Perceiver Input Channel Size	Image: 371 Audio: 81 Text: 300	Image: 35 Audio: 74 Text: 300	Static: 1 Time-series: 12	Colorless Image: 16 (cut into 4×4 squares) Audio Spectogram: 256 (cut into 16×16 squares)
	Perceiver Input Extra Axis	Image: 1 Audio: 1 Text: 1	Image: 1 Audio: 1 Text: 1	Static: 1 Time-series: 1	Colorless Image: 2 Audio Spectogram: 2
	Perceiver Input num_freq_bands	Image: 3 Audio: 3 Text: 3	Image: 3 Audio: 3 Text: 3	Static: 6 Time-series: 3	Colorless Image: 6 Audio Spectogram: 6
	Perceiver Input max_freq	Image: 1 Audio: 1 Text: 1	Image: 1 Audio: 1 Text: 1	Static: 1 Time-series: 1	Colorless Image: 1 Audio Spectogram: 1