Appendix for EMR-MERGING

A Algorithm flow of EMR-MERGING

We summarize the procedure of EMR-MERGING in Algorithm 1.

Algorithm 1 EMR-MERGING Procedure

Input: Finetuned models $W_{1..N}$, pretrained model W_{pre}
Output: Unified task vector τ_{uni}, task-specific masks $M_{1..N}$, task-specific rescalers $\lambda_{1..N}$
for t **in** $1, ..., N$ **do**
  ▷ Create task vectors.
  $\tau_t = W_t - W_{pre}$
end
▷ Step 1: Elect the unified task vector.
$\gamma_{uni} = sgn(\sum_{t=1}^n \tau_t)$
$\epsilon_{uni} = zeros(d)$
for t **in** $1, ..., N$ **do**
  for p **in** $1, ..., d$ **do**
    if $\gamma_{uni}^p \cdot \tau_t^p > 0$ **then**
      $\epsilon_{uni}^p = max\,(\epsilon_{uni}^p, abs\,(\gamma_{uni}^p))$
    end
  end
end
$\tau_{uni} = \gamma_{uni} \odot \epsilon_{uni}.$
for t **in** $1, ..., N$ **do**
  ▷ Step 2: Generate task-specific masks.
  for p **in** $1, ..., d$ **do**
    $M_t^p = bool(\tau_t^p \odot \tau_{uni}^p > 0)$
  end
  ▷ Step 3: Generate task-specific rescalers.
  $\lambda_t = \frac{sum(abs(\tau_t))}{sum(abs(M_t \cdot \tau_{uni}))}$
end

B Theoretical analyses

In Section 3, we claimed that the task-specific modulators can lower the distance between the merged model and task-specific models. Here we provide detailed theoretical analyses.

Our goal is to merge model weights $W_{1..N}$ by minimizing the distance between the merged model W_{uni} and each individual models W_i, $i \in [1..N]$ **without** using any dataset $[X_i, Y_i]$, where the distance can be calculated by:

$$Dis = \frac{\sum_{i=1}^N \|W_i - W_{uni}\|^2}{N} \qquad (6)$$

The premise of merging is that all the models are fine-tuned from the same pre-trained model. Thus, Eq. 6 can be re-written:

$$Dis = \frac{\sum_{i=1}^N \|\tau_i - \tau_{uni}\|^2}{N} \qquad (7)$$

where τ_i refers to the task vector for Task i. τ_{uni} is the merged task vector. We demonstrate the effectiveness of the task-specific modulators by step.

Analysis 1: Effectiveness of Masks. Suppose we apply a mask M_i to the unified model τ_{uni} to disable elements in τ_{uni} that have the opposite sign of the corresponding elements in τ_{uni}, which can be written as:

16
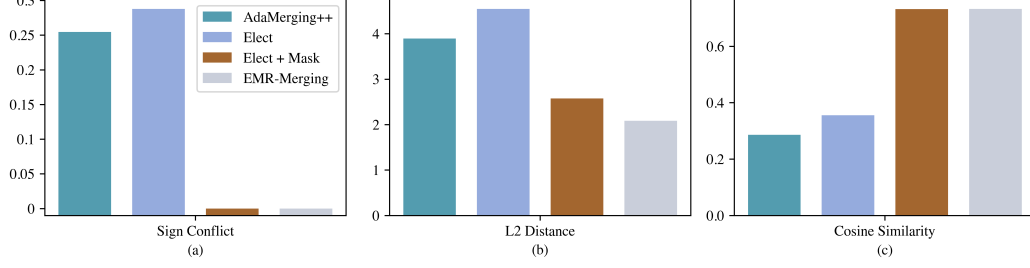
Figure 7: Comparison of (a) sign conflicts, (b) L2 distance, and (c) cosine similarity of model weights obtained by different methods (including AdaMerging++ and each procedure of EMR-MERGING) and task-specific model weights. The detailed configuration is shown in Appendix F.

Table 11: Multi-task performance when merging ViT-B/16 models on eight tasks.

Methods	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg Acc
Task Arithmetic [30]	61.1	65.9	74.0	76.2	88.0	73.9	98.4	53.0	73.8
Ties-Merging [84]	69.1	72.5	80.5	84.0	85.0	71.5	98.1	54.9	77.0
AdaMerging [85]	70.2	80.7	81.6	94.8	91.6	95.8	98.5	66.2	84.9
AdaMerging++ [85]	71.8	80.8	84.1	94.3	91.9	94.5	98.7	69.8	85.7
EMR-MERGING (Ours)	**78.6**	**82.6**	**95.5**	**99.2**	**97.6**	**98.8**	**99.6**	**78.3**	**91.3**

$$M_i = (\tau_i \odot \tau_{uni} > 0) \tag{8}$$

By applying the masks $M_i, i \in [1..N]$, the distance becomes:

$$Dis^M = \frac{\sum_{i=1}^{N} \|\tau_i - M_i \odot \tau_{uni}\|^2}{N} \tag{9}$$

Furthermore, it can be written as:

$$
\begin{aligned}
Dis^M &= \frac{\sum_{i=1}^{N} \|M_i \odot \tau_i - M_i \odot \tau_{uni}\|^2}{N} + \frac{\sum_{i=1}^{N} \|(1 - M_i) \odot \tau_i\|^2}{N} \\
&= \frac{\sum_{i=1}^{N} \|M_i \odot (abs(\tau_i) - abs(\tau_{uni}))\|^2}{N} + \frac{\sum_{i=1}^{N} \|(1 - M_i) \odot abs(\tau_i)\|^2}{N}
\end{aligned} \tag{10}
$$

where $abs(\cdot)$ returns the absolute value of each element in the input. For ease of comparison, the distance without applying M_i can be formulated as:

$$
\begin{aligned}
Dis &= \frac{\sum_{i=1}^{N} \|M_i \odot (abs(\tau_i) - abs(\tau_{uni}))\|^2}{N} + \frac{\sum_{i=1}^{N} \|(1 - M_i) \odot (abs(\tau_i) + abs(\tau_{uni}))\|^2}{N} \\
&= Dis^M + \frac{\sum_{i=1}^{N} \|(1 - M_i) \odot abs(\tau_{uni})\|^2}{N}
\end{aligned} \tag{11}
$$

Thus, we demonstrate that $Dis^M \leq Dis$, indicating applying task-specific masks can reduce the distance between the merged model and individual models, thus showing effectiveness.

Analysis 2: Effectiveness of Rescalers. Suppose we apply a rescaler $\lambda_i > 0$ to the masked unified task vector $M_i \cdot \tau_{uni}$, the distance becomes:

$$
\begin{aligned}
Dis^{M,\lambda} &= \frac{\sum_{i=1}^{N} \|\tau_i - \lambda_i \cdot M_i \odot \tau_{uni}\|^2}{N} \\
&= \frac{\sum_{i=1}^{N} \|abs(\tau_i) - \lambda_i \cdot abs(M_i \odot \tau_{uni})\|^2}{N}
\end{aligned} \tag{12}
$$

17

Figure 8: t-SNE visualization results of different merging methods.

Table 12: Multi-task performance when merging ViT-B/32 models on 9 vision tasks (ImageNet-1K added).

Methods	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	ImageNet-1K	Avg Acc
Individual	75.3	77.7	96.1	99.7	97.5	98.7	99.7	79.4	82.0	89.6
Weight Averaging	61.8	56.4	65.9	66.2	62.7	44.5	81.8	49.0	61.5	61.1
Task Arithmetic [30]	51.8	30.9	55.8	64.3	69.0	42.2	92.7	46.8	66.6	57.8
Ties-Merging [84]	53.3	34.1	57.0	55.8	72.3	43.2	90.5	46.5	68.9	58.0
EMR-MERGING (Ours)	**77.0**	**75.2**	**92.9**	**92.7**	**79.7**	**90.2**	**97.6**	**76.2**	**79.8**	**84.6**

To minimize the distance in Eq. 12, we set the first derivative of Dis^{λ} with respect to λ_i to 0, thus λ_i can be calculated by:

$$\lambda_i = \frac{sum(abs(\tau_i))}{sum(abs(M_i \odot \tau_{uni}))} \tag{13}$$

which exactly matches our setting of λ_i. This indicates that our setting of rescalers λ_i can minimize the distance between the merged model and individual models, which is: $Dis^{M,\lambda} \leq Dis^M$, thus showing effectiveness.

It is also reflected in Fig. 7 that after Masking and Rescaling, the sign conflicts and L2 distance between the merged model and task-specific models are reduced and the cosine similarity can is improved.

18
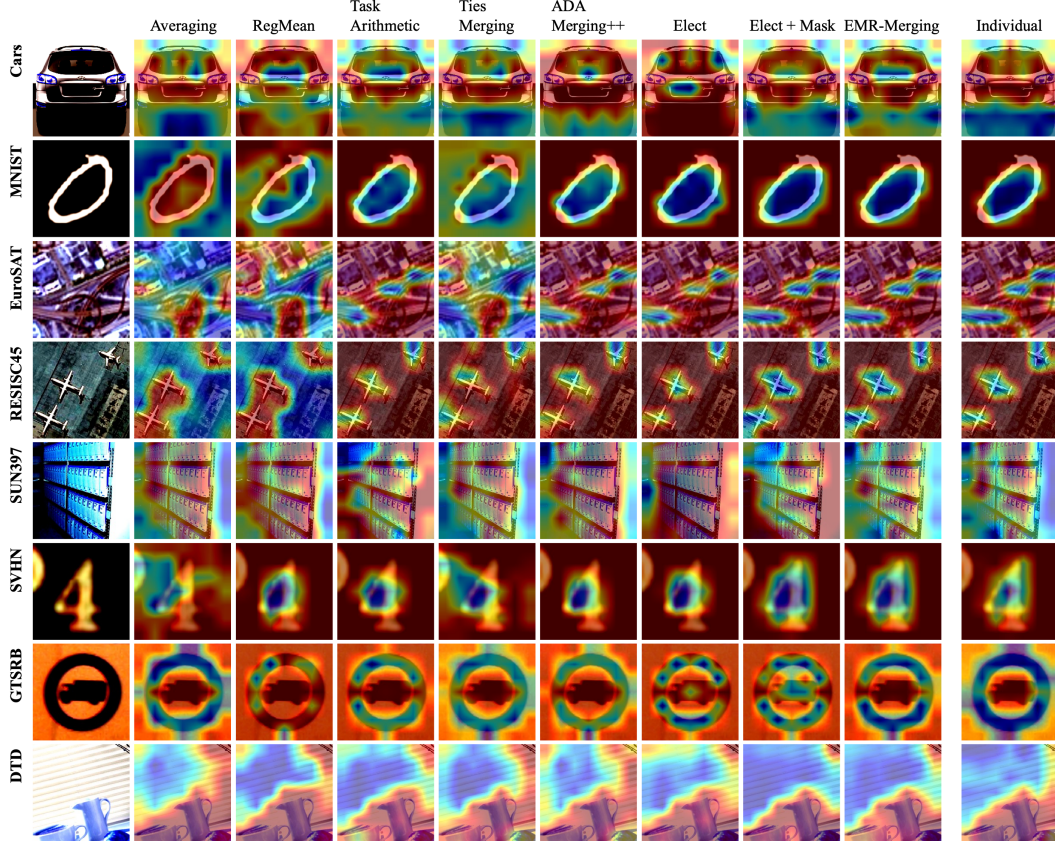
Figure 9: Grad-CAM visualization results of different merging methods.

C Baseline Methods

- **Individual Models** refer to task-specific models before merging.

- **Traditional MTL** uses datasets from all the tasks to train a single model jointly.

- **Weight Averaging** element-wisely averages all the model weights. Its effectiveness when applied to fine-tuned model weights from the same pre-training has been verified [80, 57, 33].

- **Fisher Merging** [46] uses Fisher information matrices [23] to calculate the importance of each parameter and weighted merges them based on their importance.

- **RegMean** [33] weighted merges models based on a closed-form solution to the merging problem. When merging K linear model weights W_i, where $f_i(x) = W_i^T x$, $i = 1..K$, the merging problem can be formulated as: $\min_W \sum_{i=1}^{K} \|W^T X_i - W_i^T X_i\|^2$, where W is the merged model weights, and X_i denotes the input of ith model. The closed-form solution to the problem is: $W = (\sum_{i=1}^{K} X_i^T X_i)^{-1}(\sum_{i=1}^{K} X_i^T X_i W_i)$. Inner-product matrices need to be computed before merging.

- **Task Arithmetic** [30] defines task vectors as the difference between finetuned model weights and the pre-trained model weights. Suppose a model θ_i is finetuned from a pre-trained model θ_{pre}, the task vector is $\tau_i = \theta_i - \theta_{pre}$. When merging $\theta_{1..K}$, the merged model is $\theta_M = \lambda \sum_{i=1}^{K} \tau_i + \theta_{pre}$, where λ is the merging coefficient.

- **Ties-Merging** [84] (Trim, Elect Sign & Merge) believes that the conflicts among the task vectors severely effect the merged model's performance. Ties-Merging solves this problem by eliminating redundant parameters and resolving symbol conflicts.

19

Table 13: Performance of RegMean and Task Arithmetic when pre-processed using DARE [90].

| Methods | Single-Sentence Tasks | | Similarity and Paraphrase Tasks | | | Inference Tasks | | |
	CoLA	SST2	MRPC	STSB	QQP	MNLI	QNLI	RTE
Individual	0.6018	0.9404	0.8922	0.9063	0.9141	0.8720	0.9271	0.7906
EMR-MERGING (Ours)	0.3996	0.9335	0.8627	0.8277	0.8972	0.8545	0.8957	0.7437
RegMean [33]	0.3667	0.906	0.7574	0.6268	0.8355	0.7002	0.8235	0.5848
w/ DARE (drop 10%)	0.5046	0.5298	0.3603	0.1533	0.4955	0.3245	0.4924	0.4477
w/ DARE (drop 30%)	0.4535	0.6135	0.3186	0.0471	0.4219	0.3325	0.505	0.5126
w/ DARE (drop 50%)	0.2758	0.5138	0.3211	-0.0965	0.3685	0.3338	0.508	0.5235
w/ DARE (drop 70%)	0	0.4908	0.3162	0.0021	0.3682	0.3184	0.5056	0.4838
w/ DARE (drop 90%)	0	0.4908	0.3162	-0.0776	0.3682	0.3187	0.5158	0.4910
Task Arithmetic [30]	0.1878	0.8589	0.7990	0.7403	0.8378	0.5908	0.6967	0.6209
w/ DARE (drop 10%)	0.2424	0.8509	0.7966	0.7234	0.8382	0.5869	0.7368	0.6101
w/ DARE (drop 30%)	0.3040	0.8452	0.7941	0.6311	0.8333	0.5515	0.786	0.6137
w/ DARE (drop 50%)	0.2451	0.8188	0.7990	0.4262	0.8099	0.4591	0.7269	0.6029
w/ DARE (drop 70%)	0	0.7225	0.6373	0.1353	0.7321	0.3453	0.6495	0.5162
w/ DARE (drop 90%)	0	0.4908	0.3162	0.0422	0.3682	0.3185	0.5114	0.4729
Ties-Merging [84]	0.2048	0.8440	0.8113	0.5819	0.8570	0.6465	0.7481	0.4296
w/ DARE (drop 30%)	0	0.5103	0.3382	-0.0024	0.3961	0.3238	0.5277	0.4838
w/ DARE (drop 50%)	0.0464	0.6021	0.5343	0.0192	0.6846	0.3410	0.5841	0.4982
w/ DARE (drop 70%)	0.1342	0.7833	0.7672	0.1667	0.8180	0.4172	0.691	0.5271
w/ DARE (drop 90%)	0.2618	0.8383	0.8039	0.6082	0.8336	0.5551	0.7692	0.5235

- **AdaMerging** [85] uses an unsupervised method to learn the merging coefficients for each task vector (Task-wise AdaMerging) or each layer (Layer-wise AdaMerging). AdaMerging++ is realized by adopting Ties-Merging [84] before learning the merging coefficients.

- **DARE** [90] (Drop and Rescale) validates the extremely redundant properties of language models. As a pre-processing technique, DARE randomly drops most (90% or even 99%) delta parameters (task vectors) before merging to potentially mitigate the interference of parameters among models.

D More experimental results

D.1 Merging ViT-B/16 models on 8 tasks

We follow the settings in Section 4.1.1 and merge ViT-B/16 models. Tab. 11 shows the accuracy of merging ViT-B/16 models on eight vision tasks. The proposed EMR-MERGING brings about 5.6% performance improvement compared to Adamerging++ [85], further demonstrating the effectiveness of EMR-MERGING.

D.2 Merging ViT-B/32 models on 9 tasks (ImageNet-1K added)

To further explore the performance of EMR-MERGING, we follow the settings in Section 4.1.1 and add one more task, ImageNet-1K [18]. We merge models on these nine tasks using different merging methods. The results are shown in Tab. 12 and EMR-Merging shows a much more significant improvement compared to existing merging methods (up to 20%).

D.3 DARE's experimental results and causes

DARE's experimental results when combined with RegMean and Task Arithmetic are shown in Tab. 13. It can be seen that when applied to merge eight models, DARE works on a few tasks under low dropping rate settings but it generally fails. We attribute its failure to the random dropping strategy's unapplicability to merging multiple models. Under the setting of merging two or three models, randomly dropping most parameters in task vectors can significantly reduce interference but conflicts are a lot more difficult to avoid when merging multiple models.

20

Table 14: Performance of Task Arithmetic [30], Ties-Merging [84], Ties-Merging [84] w/ DARE [90], and RegMean [33] under different hyper-parameter settings. λ for task vector-based methods is the merging coefficient. P is the drop rate for DARE. a is the non-diagonal multiplier for RegMean.

Methods	Single-Sentence Tasks		Similarity and Paraphrase Tasks			Inference Tasks		
	CoLA	**SST2**	**MRPC**	**STSB**	**QQP**	**MNLI**	**QNLI**	**RTE**
Individual	0.6018	0.9404	0.8922	0.9063	0.9141	0.872	0.9271	0.7906
EMR-MERGING (Ours)								
	0.3996	0.9335	0.8627	0.8277	0.8972	0.8545	0.8957	0.7437
Task Arithmetic								
$\lambda = 0.1$	0.0464	0.742	0.6691	0.2344	0.771	0.3567	0.6919	0.556
$\lambda = 0.3$	0.1878	0.8589	0.799	0.7403	0.8378	0.5908	0.6967	0.6209
$\lambda = 0.5$	-0.0089	0.7913	0.7794	0.5686	0.8271	0.4631	0.5387	0.4693
$\lambda = 0.7$	-0.0079	0.6525	0.7819	0.1292	0.8146	0.3949	0.5279	0.5054
$\lambda = 0.9$	-0.0207	0.7202	0.4167	-0.1283	0.8012	0.2913	0.5294	0.5162
$\lambda = 1.0$	0	0.5619	0.3554	-0.2496	0.7939	0.259	0.5338	0.5162
Ties-Merging								
$\lambda = 0.1$	0	0.4908	0.3162	0.0214	0.3682	0.3186	0.5105	0.4729
$\lambda = 0.3$	0	0.5631	0.5049	-0.0074	0.4696	0.35	0.5649	0.4621
$\lambda = 0.5$	0.2232	0.7592	0.7696	0.1149	0.827	0.4486	0.6939	0.4368
$\lambda = 0.7$	0.2507	0.8291	0.7917	0.3774	0.8488	0.5858	0.7507	0.4188
$\lambda = 0.9$	0.2048	0.844	0.8113	0.5819	0.857	0.6465	0.7481	0.4296
$\lambda = 1.0$	0.1712	0.8406	0.799	0.6444	0.859	0.6409	0.7069	0.426
Ties-Merging w/ DARE								
$\lambda = 0.2, P = 0.3$	0	0.4920	0.3162	0.0053	0.3682	0.3186	0.5131	0.4477
$\lambda = 0.2, P = 0.5$	0	0.0043	0.3162	0.0036	0.3690	0.3202	0.5226	0.4946
$\lambda = 0.2, P = 0.7$	0.0464	0.6388	0.5735	0.0301	0.0047	0.3383	0.5984	0.5090
$\lambda = 0.2, P = 0.9$	0.2402	0.8165	0.7843	0.2696	0.8112	0.4384	0.7223	0.5415
$\lambda = 0.3, P = 0.3$	0	0.5103	0.3382	-0.0024	0.3961	0.3238	0.5277	0.4838
$\lambda = 0.3, P = 0.5$	0.0464	0.6021	0.5343	0.0192	0.6846	0.3410	0.5841	0.4982
$\lambda = 0.3, P = 0.7$	0.1342	0.7833	0.7672	0.1667	0.8180	0.4172	0.691	0.5271
$\lambda = 0.3, P = 0.9$	0.2618	0.8383	0.8039	0.6082	0.8336	0.5551	0.7692	0.5235
$\lambda = 0.4, P = 0.3$	0.0656	0.6216	0.5588	0.0192	0.7301	0.3461	0.5891	0.5162
$\lambda = 0.4, P = 0.5$	0.1172	0.7374	0.7451	0.1045	0.8157	0.3913	0.6667	0.5126
$\lambda = 0.4, P = 0.7$	0.2440	0.8234	0.7843	0.3955	0.8371	0.5496	0.7216	0.4838
$\lambda = 0.4, P = 0.9$	0.1380	0.8440	0.8064	0.7044	0.8365	0.5835	0.6529	0.5054
RegMean								
$a = 0.7$	0.3005	0.9037	0.7525	0.6349	0.8322	0.6794	0.8157	0.5632
$a = 0.8$	0.3346	0.9014	0.7549	0.6375	0.8339	0.6841	0.8173	0.5704
$a = 0.9$	0.3445	0.9048	0.7525	0.6362	0.8361	0.6918	0.821	0.5632
$a = 1.0$	0.3667	0.906	0.7574	0.6268	0.8355	0.7002	0.8235	0.5848

D.4 Results under different hyper-paramerter settings

In Section 4.2.1, we presented the best performance of Ties-Merging, Task Arithmetic, and RegMean among multiple hyper-parameter settings. Here we present more experimental results of Ties-Merging, Task Arithmetic, and RegMean under different hyper-parameter settings in Tab. 14.

D.5 Detailed information for merging different number of models

In Section 4.4, we showed partial results of merging different number of ViT-B/32 models by Fig. 6. Here we provide quantified and task-specific performance results in Tab. 15.

21

Table 15: Merging different number of ViT-B/32 models.

Methods	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg Acc
Individual									
2 Tasks	75.3	77.7	-	-	-	-	-	-	76.5
3 Tasks	75.3	77.7	96.1	-	-	-	-	-	83.0
4 Tasks	75.3	77.7	96.1	99.7	-	-	-	-	87.2
5 Tasks	75.3	77.7	96.1	99.7	97.5	-	-	-	89.3
6 Tasks	75.3	77.7	96.1	99.7	97.5	98.7	-	-	90.8
7 Tasks	75.3	77.7	96.1	99.7	97.5	98.7	99.7	-	92.1
8 Tasks	75.3	77.7	96.1	99.7	97.5	98.7	99.7	79.4	90.5
Ties-Merging									
2 Tasks	69.2	68.2	-	-	-	-	-	-	68.7
3 Tasks	69.2	68.0	78.9	-	-	-	-	-	72.0
4 Tasks	68.9	67.9	79.4	86.0	-	-	-	-	75.5
5 Tasks	68.6	67.1	79.0	83.5	66.6	-	-	-	73.0
6 Tasks	68.0	66.4	77.9	80.1	74.4	69.9	-	-	72.8
7 Tasks	66.6	65.7	75.7	76.7	81.0	69.2	96.4	-	75.9
8 Tasks	64.8	62.9	74.3	78.9	83.1	71.4	97.6	56.2	72.4
EMR-MERGING (Ours)									
2 Tasks	78.9	76.1	-	-	-	-	-	-	77.5
3 Tasks	77.9	75.2	95.3	-	-	-	-	-	82.8
4 Tasks	77.4	74.9	94.8	99.7	-	-	-	-	86.7
5 Tasks	77.2	74.2	94.7	99.7	97.1	-	-	-	88.6
6 Tasks	76.4	73.4	94.2	99.7	97.0	98.5	-	-	89.9
7 Tasks	75.8	73.3	93.6	99.6	96.9	98.2	99.6	-	91.0
8 Tasks	75.2	72.8	93.5	99.5	96.9	98.1	99.6	74.4	88.7

Table 16: Sparsity (ratio of non-zero items) of the masks and the values of the rescalers when merging ViTs on 8 vision tasks and RoBERTa models on 8 language tasks.

Sparsity	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD
ViT-B/32	0.7194	0.7121	0.7106	0.6994	0.7195	0.7062	0.7132	0.7058
ViT-L/14	0.6832	0.6699	0.6734	0.6579	0.6748	0.6444	0.6614	0.6620

Rescalers	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD
ViT-B/32	0.7489	0.7635	0.7489	0.7476	0.7962	0.7652	0.7981	0.7624
ViT-L/14	0.7656	0.7652	0.7537	0.7384	0.7874	0.7313	0.7763	0.7638

Sparsity	CoLA	SST2	MRPC	STSB	QQP	MNLI	QNLI	RTE
RoBERTa	0.6264	0.6547	0.6498	0.6150	0.7620	0.7739	0.6243	0.5979

Rescalers	CoLA	SST2	MRPC	STSB	QQP	MNLI	QNLI	RTE
RoBERTa	0.2458	0.4698	0.5033	0.2078	0.8891	0.8987	0.4683	0.1466

D.6 Sparsity of masks and values of rescalers.

We show the sparsity of the masks and the values of the rescalers when merging eight ViTs and eight RoBERTa models in Tab. 16.

E More visualization results

In Section 3, we showed some visualization results using t-SNE [69] and Grad-CAM [61]. Here we provide more visualization results of both existing merging methods and EMR-MERGING. t-SNE and Grad-CAM visualization results are shown in Fig. 8 and Fig. 9, respectively.

22

F Configuration of Fig. 4 and Fig. 7

In Fig. 4 and Fig. 7, we hope to compare the sign conflicts, L2 distance, and cosine similarity of the merged model weights and individual model weights. To calculate the sign conflicts, we element-wisely compare the merged model weights to each individual model weights and record the ratio of the elements whose signs conflict. We report the average value of the sign conflicts between the merged model and each individual model. To calculate the L2 distance or cosine similarity, we first flatten the merged model weights and each individual model weights as 1-dimension vectors. Then we calculate the L2 distance or cosine similarity between the merged model and each individual model and report the average value.

G Limitations and future works

Despite the convincing results, the proposed method suffers from several limitations. On the one hand, compared to existing methods, EMR-MERGING requires a little additional memory to store the light-weight task-specific modulators. On the other hand, as a common limitation of task vector-based methods, EMR-MERGING cannot be generalized to models trained from-scratch because the task vector is based on the pretrain-finetune paradigm.

Further improving the performance of the merged model and generalizing model merging to models trained from-scratch or even models with different structures are significant directions for future work. Additionally, combining model merging with low bit-width quantization has broad application prospects and is also a potential future work.

23