Less is More: Efficient Model Merging with Binary Task Switch

Supplementary Material

7. Exprimental Details

To provide a comprehensive overview of the experimental setup, we list the hyperparameter settings for our method and all baseline methods in Table 5.

Methods	α	N	Scaling Coef	LR	Epochs
Task-Arithmetic	–	–	0.3	–	–
Ties-Merging	0.5	–	0.3	–	–
DARE	0.5	–	–	–	–
RegMean	–	256	–	–	–
Fisher-Merging	–	4096	–	–	–
AdaMerging	–	–	–	1e-3	500
AdaMerging++	0.5	–	–	1e-3	500
Twin-Merging	0.5	100	0.3	1e-3	10
EMR-Merging	–	–	–	–	–
T-Switch(Ours)	0.5	–	–	–	–
Auto-Switch(Ours)	0.5	100	–	–	–

Table 5. Hyperparameter settings of our method and all baselines.

Here, α represents the discard rate of task vector parameters, N denotes the number of example samples retained for each task, and "Scaling Coef" refers to the scaling coefficient applied to the merged task vector. LR indicates the learning rate used for training the merging weights or coefficients (e.g., AdaMerging and AdaMerging++) or the task router (e.g., Twin-Merging). "Epochs" refers to the additional training epochs required for the merging method. A dash ('-') in the table indicates that the corresponding method does not involve the specified hyperparameter. The hyperparameter settings for all baseline methods follow the configurations provided in the original papers.

In the low rank experiment, we set the rank of LoRA to 64 and only added it on conv1 and mlp, and trained 20 epochs on different datasets to obtain corresponding low-rank task vectors.

For all methods that require setting discard ratio, we searched between 0.1-0.9 and selected the optimal discard ratio for different methods. The Table 6 shows the optimal discard ratios for different methods.

Method	ViT-B/32	ViT-B/32+LoRA	ViT-L/14	RoBERTa
DARE	0.6	0.6	0.7	0.1
Adamerging++	0.7	0.5	0.4	–
Ties-Merging	0.3	0.9	0.3	0.9
T-Switch (ours)	0.7	0.5	0.6	0.4
Auto-Switch (ours)	0.6	0.6	0.5	0.4

Table 6. The optimal discard ratio for different baselines.

8. Additional Results on ViT models

8.1. Merging results on the ViT-L/14 model

To evaluate the effectiveness of our method in merging larger models, we conducted experiments on eight visual tasks using the ViT-L/14 model. Table 7 shows the combined performance of our method and various baseline methods. Our Auto-Switch achieved the best results, significantly outperforming other baseline methods. Notably, T-Switch even surpasses the average performance achieved in the Individual case. This demonstrates that our T-Switch retains excellent performance on larger visual models and validates the generalizability of our proposed methods.

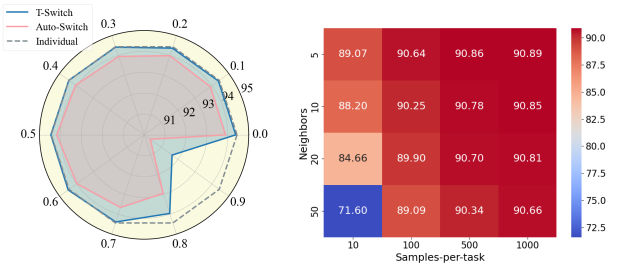

Figure 7. Additional ablation results. Left: Merging results(%) with discard ratios ranging from 0.0 to 0.9 on the ViT-L-14 model. Right: Ablation results of the Auto-Switch method merging eight visual tasks on the ViT-B/32 model when the discard ratio is 0.5.

Additionally, we have verified the impact of different discard rates α on the merging performance of our method on ViT-L by conducting ablation experiments on the ViT-L/14 model with various discard rates α. The left panel of Fig.7 presents the ablation results for merging eight visual tasks with discard ratio α ranging from 0.0 to 0.9 using the ViT-L/14 model. From the results, we observe a similar phenomenon to that found in the main experiments: as the discard ratio increases, the performance of T-Switch improves, even surpassing fine-tuning performance at a discard ratio of 0.6, with noticeable performance degradation only occurring beyond $\alpha = 0.7$. As the number of discarded redundant parameters increases, the interference noise in the task vector will also decrease. Consequently, the performance of T-Switch and Auto-Switch shows improvement with a moderate increase in α.

Type	Methods	Automatic	Example	Storage	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	AVG
–	Pretrained	–	–	–	66.87	77.94	71.33	62.22	58.45	50.55	76.36	55.37	64.89
	Traditional MTL	–	–	–	80.80	90.60	96.30	96.30	97.60	99.10	99.60	84.40	93.09
	Individual	–	–	–	84.86	92.39	97.37	99.74	98.11	99.24	99.69	84.15	94.44
	T-Switch(Ours)	–	–	–	84.75	92.61	97.38	99.74	98.12	99.25	99.73	84.31	94.49
Static	Weight-Averaging	–	✗	–	71.10	81.56	82.60	90.63	78.23	70.65	97.01	62.77	79.32
	Task-Arithmetic	–	✗	–	73.91	82.13	86.65	92.70	87.91	86.78	98.94	65.64	84.33
	Ties-Merging	–	✗	–	73.95	80.27	85.85	91.67	90.50	88.03	99.67	65.73	84.46
	DARE	–	✗	–	73.80	82.51	86.87	93.12	88.06	87.36	98.92	66.07	84.59
	RegMean	–	✓	–	73.04	86.10	88.40	97.52	91.53	89.78	99.0	69.95	86.91
	Fisher-Merging	–	✓	–	68.11	84.54	75.13	84.11	95.64	91.36	95.56	67.23	82.71
	AdaMerging	–	✓	–	79.00	90.30	90.80	96.20	93.40	98.00	99.00	79.90	90.83
	AdaMerging++	–	✓	–	79.53	90.75	91.47	96.53	94.05	98.02	99.02	80.80	91.27
Dynamic	Twin-Merging	✓	✓	10476.1	**84.41**	91.57	**96.95**	**99.70**	**98.18**	92.40	99.74	**84.52**	93.43
	EMR-Merging	✗	✗	1391.5	83.17	90.71	96.78	**99.70**	97.94	99.09	99.69	82.71	93.73
	Auto-Switch(Ours)	✓	✓	174.4	83.27	**92.50**	96.71	99.67	98.12	**99.24**	**99.75**	84.15	**94.18**

Table 7. Main results of merging full-rank task vectors of the ViT-L/14 model on eight vision datasets. The best method is highlighted in bold, and the second-best method is underlined.

8.2. Ablation of Auto-Switch hyperparameters: samples-per-task N and number of neighbors C

In our proposed Auto-Switch, there are two additional hyperparameters of this method besides the discard ratio α, namely the number of samples retained for each task N and the number of neighbors C. To verify the impact of these two hyperparameters on Auto-Switch, we conduct ablation experiments on the ViT-B/32 model. The results shown in Fig.7 indicates that: (1) As the number of neighbors C increases, the model's merging performance tends to decrease. This happens because, when selecting neighbors from the local vicinity of the input sample, an increase in the number of neighbors (especially when it approaches the total number of samples) can introduce many distant, irrelevant samples. These distant samples can negatively impact classification accuracy, thereby reducing the effectiveness of the merging process. Therefore, selecting an appropriate number of neighbors C, is crucial. (2) As the number of samples per task N increases, the model's merging performance improves significantly. This is because more samples help to concentrate the features of each dataset, which in turn enhances the stability of neighbor selection.

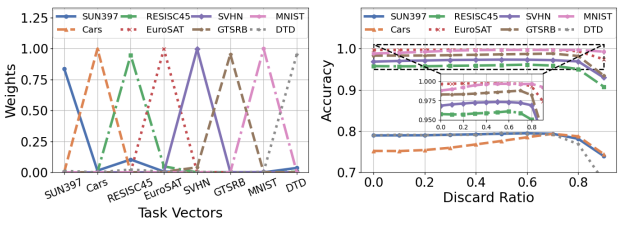

Figure 8. Left: The weights assigned to each binary task vectors; Right: Performance across various tasks as the discard ratio varies.

8.3. Exploring the weight allocation of Auto-Switch to different tasks

We validated it using the ViT-B/32 model on different task test sets to further investigate the distribution of weight allocation in Auto-Switch. The left panel of Fig.8 shows the weight allocation to all task vectors when evaluating a certain task. Auto-Switch allocate most weight to the task vector related to the task, but regardless of the number of models merged, only one merging process is required.

8.4. The impact of Bin-Discard's discard ratio variation on each task

We tested the impact of varying discard rates on each task, with results for the ViT-B/32 model shown in the right panel of Fig.8. The trends were consistent across tasks: initially increasing, then decreasing, with optimal performance observed at discard rates between 0.6 and 0.7.

Methods	ViT-B/32	ViT-L/14	Roberta
base-model	113.595	868.488	75.317
Twin-Merging	158.72	1851.57	211.09
Auto-Switch	161.98	1854.78	214.47

Table 8. The total time (s) required to evaluate the test set for all tasks.

8.5. Analysis of inference speed in Auto-Switch

We have added analysis on inference speed for better understanding on Auto-Switch. As shown in the Table 8, on an Nvidia A100 GPU with a ViT-B/32 base model, Twin-Merging achieves 71.57% of direct inference speed, while our Auto-Switch reaches 72.01%, but with only 1.66% of Twin-Merging's storage. This demonstrates the excellent trade-off between inference complexity and storage efficiency of our approach.

9. Additional Results on LLMs

To extensively validate the effectiveness of the proposed method, we perform merging using our methods and several baselines on two models with the same architecture but fine-tuned in different domains: WizardMath-13B (Math) and llama-2-13b-codealpaca (Code). Since the fine-tuning data for these two LLMs is not available, some baselines such as RegMean, Fisher-Merging, Ada-Merging, Ada-Merging++, and Twin-Merging could not be reproduced. It is important to note that in our Auto-Switch method, we replace KNN with Sentence Transformer to calculate semantic similarity, and use it to assign weights to task vectors from different domains. The merging results are shown in the Table 9.

Methods	GSM8K	MATH	HumanEval	MBPP	AVG
WizardMath-13B	63.53	14.14	7.32	19.60	26.15
T-Switch (Math)	63.91	13.72	6.78	19.46	25.97
llama-2-13b-codealpaca	0.00	0.00	25.00	27.80	13.20
T-Switch (Code)	0.00	0.00	23.17	27.60	12.69
Weight-Averaging	55.88	10.50	8.54	8.40	20.83
Task-Arithmetic	<u>63.59</u>	**14.01**	9.76	8.80	24.04
DARE	**63.97**	<u>13.99</u>	11.59	10.00	24.89
Ties-Merging	62.54	13.68	10.98	22.60	27.45
EMR-Merging	63.28	13.08	<u>20.12</u>	<u>26.20</u>	<u>30.67</u>
Auto-Switch (Ours)	63.46	13.64	**23.78**	**27.60**	**32.12**

Table 9. The merging results of the LLMs, evaluated using the pass@1 metric for both HumanEval and MBPP. The result of DARE is based on the Task-Arithmetic method. The best method is highlighted in bold, and the second-best method is underlined.

In the Task-Arithmetic baseline, since the task vectors from the WizardMath-13B model are longer than those from the llama-2-13b-codealpaca model, the merged model tends to retain more mathematical abilities. As a result, the model becomes proficient in mathematical tasks but loses a significant amount of code-related knowledge.

As concluded in the main paper, our proposed methods still demonstrate superior performance in the merging of LLMs. Surprisingly, the approximate model obtained using T-Switch even outperforms WizardMath-13B on the GSM8K. As for the other three test sets, there is a slight performance decline, which is reasonable because the binarization operation in T-Switch discards a significant amount of information from the task vectors. Meanwhile, the Auto-Switch outperforms the T-Switch on the HumanEval, which to some extent proves that there may be useful information within task vectors from different domains that aids the current task.