DeRS: Towards Extremely Efficient Upcycled Mixture-of-Experts Models

Supplementary Material

A. Extended DeRS Compression and Upcycling

In our medical multi-modal and code generation experiments, the original FFN layer in a pre-trained dense model is upcycled into a parallel structure consisting of a universal FFN layer and a MoE layer containing N FFN experts. The universal FFN and the N experts are all initialized from the original FFN weight. The universal FFN processes all inputs, while the N experts are sparsely activated by a router for each input. The outputs from the universal FFN and the MoE layer are then summed to form the final output.

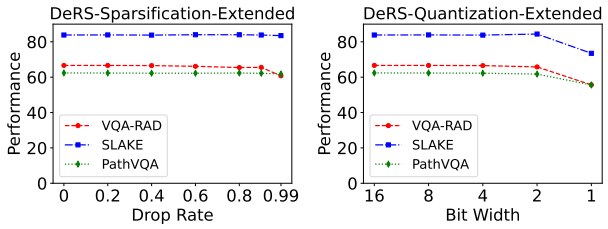
In the main body, we applied the proposed DeRS paradigm only to the N experts in the MoE layer, since the universal FFN is not sparsely activated by the router, meaning it cannot be strictly considered as a MoE expert. Here, considering that both the universal FFN and the N MoE experts share the same initial weight, we extend our DeRS compression and DeRS upcycling to the universal FFN layer to further reduce parameter redundancy.

Specifically, when applying the extended DeRS compression to compress a vanilla upcycled MoE model, both the universal FFN and the N MoE experts are treated as a whole and decomposed into one expert-shared base weight and $N+1$ delta weights. Subsequently, sparsification or quantization techniques are applied to the $N+1$ delta weights to reduce redundancy. Similarly, when applying the extended DeRS upcycling to convert a pre-trained dense model into the MoE architecture, the universal FFN and the N MoE experts are treated as a whole, sharing one base FFN and introducing $N+1$ unique, parameter-efficient weights in the form of sparse or low-rank matrixes.

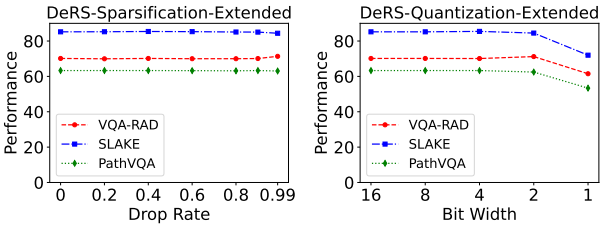
A1. Extended DeRS Compression on Medical Task

Fig. S1 shows the performance of applying the extended DeRS compression to two vanilla upcycled Med-MoE models on the medical multimodal task. The detailed results are presented in Tab. S5 and Tab. S6. As we can see, even when simultaneously compressing the universal FFN and MoE experts, the extended DeRS-Sparsification with a 0.8 drop rate and the extended DeRS-Quantization with a 4-bit width can reduce the additional parameter count by 75% and 69%, respectively, while maintaining performance.

Different from the results shown in Fig. 5, where only the MoE experts were compressed, simultaneously compressing the universal FFN and MoE experts leads to a slight performance drop under extreme compression settings (0.99 drop rate or 1-bit width). This degradation occurs because extreme compression of the universal FFN significantly impacts the model's output, as the universal FFN processes all input tokens. However, since the two dense models utilized within the Med-MoE framework have been previously fine-tuned on relevant yet non-overlapping medical multi-modal datasets, the overall performance of upcycled MoE models does not collapse under these extreme compression settings.



(a) Med-MoE-StableLM



(b) Med-MoE-Phi

Figure S1. Performance of applying the extended DeRS compression to compress two vanilla upcycled Med-MoE models respectively. For each dataset, we report the average performance of the open-set and closed-set.

A2. Extended DeRS Upcycling on Medical Task

As shown in Tab. S1, when treating the construction of the universal FFN and MoE experts as a whole, both of our extended DeRS upcycling methods achieve comparable performance to vanilla upcycling while introducing significantly fewer additional parameters. For example, when achieving the same performance on the Med-MoE-Phi architecture, our extended DeRS-SM and DeRS-LM upcycling strategies introduce only 5.18 million and 9.18 million additional parameters respectively, while vanilla upcycling introduces a massive 3.36 billion parameters. These results highlight the ability of our DeRS upcycling to achieve extremely efficient upcycled MoE models.

A3. Extended DeRS Compression on Code Task

Fig. S2 shows the performance of applying the extended DeRS compression to the vanilla upcycled Coder-MoE model on the code generation task, with detailed results presented in Tab. S3 and Tab. S4. As we can see, for the delta weights obtained by the unified decomposition of the

Table S1. Performance comparison between vanilla upcycling and our extended DeRS upcycling on two Med-MoE models on the medical multi-modal task. DeRS-SM† and DeRS-LM† denote the extended Sparse-Matrix-based and Low-rank-Matrix-based DeRS upcycling respectively. **Added Params** represents the number of additional parameters of the upcycled MoE model compared to its corresponding dense model.

MoE Model	Upcycling Method	Added Params.	VQA-RAD		SLAKE		PathVQA		Overall
			Open	Closed	Open	Closed	Open	Closed	
Med-MoE-StableLM (EMNLP 24)	Vanilla	1.66B	51.0	82.3	82.4	85.3	33.4	91.4	71.0
	DeRS-SM†	2.17M	51.2	81.3	84.5	84.4	33.6	90.9	71.0
	DeRS-LM†	5.63M	50.4	81.6	83.6	84.4	33.9	91.4	70.9
Med-MoE-Phi (EMNLP 24)	Vanilla	3.36B	55.1	85.3	84.6	85.8	35.1	91.5	72.9
	DeRS-SM†	5.18M	54.8	84.6	84.0	87.2	35.0	91.6	72.9
	DeRS-LM†	9.18M	55.3	83.8	84.3	86.5	35.6	91.9	72.9

universal FFN and MoE experts, removing 40% of their elements or quantizing them to 4 bits can effectively eliminate redundancy without degrading performance. However, since the dense model utilized for constructing Coder-MoE has not undergone any prior fine-tuning, excessive simultaneous compression of both the universal FFN and MoE experts can lead to a collapse in the performance of the vanilla upcycled Coder-MoE model.

A4. Extended DeRS Upcycling on Code Task

As shown in Tab. S2, our extended DeRS upcycling remains effective and extremely efficient on the code generation task. For example, our extended DeRS-LM upcycling strategy achieves an overall performance improvement of 0.7%, while only introducing only 11.3 million additional parameters, whereas vanilla upcycling introduces a significant 3.24 billion extra parameters. These results demonstrate that our proposed DeRS upcycling method propels upcycled MoE models towards a new level of efficiency.

B. Detailed Results of DeRS Compression

Detailed results of DeRS compression in the main body are provided, namely Tab. S7 and Tab. S8 for the general multi-modal task, Tab. S9 and Tab. S10 for the medical multi-modal task, and Tab. S11 and Tab. S12 for the code generation task.

C. Training settings

The detailed training hyper-parameters and our DeRS upcycling hyper-parameters for experiments on three tasks are provided in Tab. S13.

D. Recommended Application Choices

Based on extensive experiments, we empirically summarize recommended application choices for different scenarios. If the pre-trained dense model has undergone prior fine-tuning

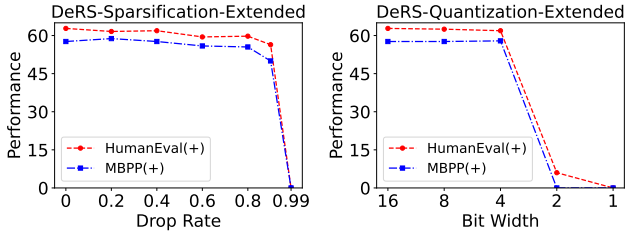


Figure S2. Performance of applying the extended DeRS compression to compress the vanilla upcycled Coder-MoE model. HumanEval(+) represents the average performance of HumanEval and HumanEval+, similarly for MBPP(+).

before upcycling, we recommend applying the sparsification-based DeRS compression to efficiently compress the vanilla upcycled MoE model, as well as utilizing sparse-matrix-based DeRS upcycling to efficiently upcycle the dense model into the MoE architecture for training. This is because, in this case, the redundancy in the delta weights is extremely high, and both sparsification and sparse matrixes can significantly reduce redundancy while maintaining performance. Conversely, if the pre-trained dense model has not undergone any prior fine-tuning, we recommend employing the quantization-based DeRS compression and the low-rank-matrix-based DeRS upcycling, as these two methods can effectively reduce redundancy while preserving global modification capabilities.

Since our proposed DeRS compression is based on the assumption that MoE experts share the same pre-trained weight initialization for the decomposition of experts and compression of redundant delta weights, it is not applicable to compressing MoE models trained from scratch. This is because training MoE models from scratch involves randomly initializing the MoE experts, making it impossible to extract redundant delta weights from the trained experts. Moreover, although our proposed DeRS upcycling has the potential to be used for training MoE models from scratch by randomly initializing the expert-shared base FFN, its performance may be limited due to insufficient model capacity.

Table S2. Performance comparison between vanilla upcycling and our extended DeRS upcycling on the code generation task. DeRS-SM† and DeRS-LM† denote the extended Sparse-Matrix-based and Low-rank-Matrix-based DeRS upcycling respectively. **Added Params** represents the number of additional parameters of the upcycled MoE model compared to its corresponding dense model.

MoE Model	Upcycling Method	Added Params.	HumanEval	HumanEval+	MBPP	MBPP+	Overall
Coder-MoE (ACL 24)	Vanilla	3.24B	64.6	61.0	63.9	51.4	60.2
	DeRS-SM†	406M	64.6	60.4	63.7	52.4	60.3
	DeRS-LM†	11.3M	65.9	62.2	63.4	51.9	60.9

Table S3. Detailed results of applying the extended DeRS-Sparsification (with different drop rates) to compress the vanilla upcycled Coder-MoE model on the code generation task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model.

Vanilla Upcycled MoE Model	Drop Rate	Added Params.	HumanEval	HumanEval+	MBPP	MBPP+
Coder-MoE (ACL 24)	0.0	3.24B	64.6	61.0	63.9	51.4
	0.2	3.24B	63.4	59.8	64.7	52.9
	0.4	2.43B	63.4	60.4	62.9	52.4
	0.6	1.62B	61.0	57.9	61.2	50.6
	0.8	0.81B	62.2	57.3	61.4	49.6
	0.9	0.41B	58.5	54.3	55.4	44.6
	0.99	0.04B	0.0	0.0	0.0	0.0

Table S4. Detailed results of applying the extended DeRS-Quantization (with different bit width) to compress the vanilla upcycled Coder-MoE model on the code generation task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model.

Vanilla Upcycled MoE Model	Bit Width	Added Params.	HumanEval	HumanEval+	MBPP	MBPP+
Coder-MoE (ACL 24)	16	3.24B	64.6	61.0	63.9	51.4
	8	2.03B	64.6	60.4	63.7	51.6
	4	1.01B	63.4	60.4	63.7	52.1
	2	0.51B	6.0	6.0	0.0	0.0
	1	0.25B	0.0	0.0	0.0	0.0

Table S5. Detailed results of applying the extended DeRS-Sparsification (with different drop rates) to compress two vanilla upcycled Med-MoE models on the medical multi-modal task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model.

Vanilla Upcycled MoE Model	**Drop Rate**	**Added Params.**	**VQA-RAD**		**SLAKE**		**PathVQA**	
			Open	Closed	Open	Closed	Open	Closed
Med-MoE-StableLM (EMNLP 24)	0.0	1.66B	51.0	82.3	82.4	85.3	33.4	91.4
	0.2	1.66B	51.0	82.3	82.5	85.3	33.3	91.4
	0.4	1.25B	50.8	82.3	82.5	85.1	33.2	91.3
	0.6	0.83B	50.3	82.0	82.4	85.6	33.1	91.4
	0.8	0.42B	48.6	82.3	82.7	85.3	33.2	91.5
	0.9	0.21B	48.8	82.3	82.4	85.3	33.0	91.4
	0.99	0.02B	42.5	79.0	81.7	85.3	32.3	91.3
Med-MoE-Phi (EMNLP 24)	0.0	3.36B	55.0	85.3	84.6	85.8	35.1	91.5
	0.2	3.36B	55.0	84.9	84.7	85.8	35.1	91.5
	0.4	2.52B	55.0	85.3	85.0	85.8	35.1	91.5
	0.6	1.68B	55.1	84.9	84.8	85.8	34.9	91.6
	0.8	0.84B	55.1	84.9	84.9	85.3	35.0	91.3
	0.9	0.42B	55.3	84.9	84.8	85.3	35.2	91.4
	0.99	0.21B	57.0	85.7	83.7	85.1	34.9	91.2

Table S6. Detailed results of applying the extended DeRS-Quantization (with different bit width) to compress two vanilla upcycled Med-MoE models on the medical multi-modal task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model.

Vanilla Upcycled MoE Model	**Bit Width**	**Added Params.**	**VQA-RAD**		**SLAKE**		**PathVQA**	
			Open	Closed	Open	Closed	Open	Closed
Med-MoE-StableLM (EMNLP 24)	16	1.66B	51.0	82.3	82.4	85.3	33.4	91.4
	8	1.04B	51.0	82.3	82.5	85.3	33.3	91.4
	4	0.52B	50.8	82.3	82.5	85.1	33.2	91.3
	2	0.26B	51.5	80.1	82.8	86.0	32.4	91.1
	1	0.13B	33.7	77.6	66.7	80.3	23.4	87.8
Med-MoE-Phi (EMNLP 24)	16	3.36B	55.0	85.3	84.6	85.8	35.1	91.5
	8	2.10B	55.0	85.3	84.6	85.8	35.1	91.5
	4	1.05B	54.9	85.3	84.9	86.0	35.1	91.5
	2	0.52B	56.7	85.7	83.7	85.3	33.5	91.4
	1	0.26B	43.6	79.4	64.2	79.8	20.1	86.6

Table S7. Detailed results of applying DeRS-Sparsification (with different drop rates) to compress three vanilla upcycled MoE-LLaVA models on the general multi-modal task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model.

Vanilla Upcycled MoE Model	Drop Rate	Added Params.	VQA^{v2}	GQA	VQA^{T}
MoE-LLaVA-StableLM (ICML 24)	0.0	1.24B	76.3	60.6	50.2
	0.2	1.33B	76.4	60.8	50.1
	0.4	1.00B	76.4	60.8	50.2
	0.6	0.66B	76.3	60.7	50.1
	0.8	0.33B	76.3	60.7	50.2
	0.9	0.17B	76.3	60.5	50.0
	0.99	0.02B	74.8	59.4	47.4
MoE-LLaVA-Qwen (ICML 24)	0.0	1.22B	76.2	61.2	48.1
	0.2	1.30B	76.2	61.3	47.7
	0.4	0.97B	76.2	61.1	48.0
	0.6	0.65B	76.2	61.3	47.5
	0.8	0.32B	76.1	61.0	47.8
	0.9	0.16B	76.1	61.1	47.5
	0.99	0.02B	73.9	59.3	42.7
MoE-LLaVA-Phi (ICML 24)	0.0	2.52B	77.5	61.4	50.8
	0.2	2.68B	77.5	61.1	50.8
	0.4	2.01B	77.5	61.1	50.9
	0.6	1.34B	77.4	61.4	50.9
	0.8	0.67B	77.5	61.4	51.0
	0.9	0.34B	77.4	61.3	50.9
	0.99	0.03B	76.9	60.6	50.2

Table S8. Detailed results of applying DeRS-Quantization (with different bit width) to compress three vanilla upcycled MoE-LLaVA models on the general multi-modal task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model.

Vanilla Upcycled MoE Model	Bit Width	Added Params.	VQA^{v2}	GQA	VQA^{T}
MoE-LLaVA-StableLM (ICML 24)	16	1.24B	76.3	60.6	50.2
	8	0.83B	76.4	60.4	50.2
	4	0.42B	76.3	60.6	50.1
	2	0.21B	76.2	60.5	50.7
	1	0.10B	74.1	55.8	48.1
MoE-LLaVA-Qwen (ICML 24)	16	1.22B	76.2	61.2	48.1
	8	0.81B	76.2	61.1	48.0
	4	0.41B	76.2	61.0	47.9
	2	0.20B	76.1	60.9	48.7
	1	0.10B	74.4	57.5	47.8
MoE-LLaVA-Phi (ICML 24)	16	2.52B	77.5	61.4	50.8
	8	1.68B	77.5	61.2	51.1
	4	0.84B	77.5	61.2	50.8
	2	0.42B	77.5	61.4	50.7
	1	0.21B	75.9	58.8	49.8

Table S9. Detailed results of applying DeRS-Sparsification (with different drop rates) to compress two vanilla upcycled Med-MoE models on the medical multi-modal task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model. The light-gray Added Params denotes the additional parameters introduced by the universal FFN layers that are not considered as experts of MoE layers.

Vanilla Upcycled MoE Model	Drop Rate	Added Params.	VQA-RAD		SLAKE		PathVQA	
			Open	Closed	Open	Closed	Open	Closed
Med-MoE-StableLM (EMNLP 24)	0.0	0.42B+1.24B	51.0	82.3	82.4	85.3	33.4	91.4
	0.2	0.42B+1.33B	50.6	82.3	82.3	85.3	33.3	91.3
	0.4	0.42B+1.00B	50.8	82.3	82.4	85.3	33.3	91.2
	0.6	0.42B+0.66B	50.6	82.3	82.4	85.3	33.2	91.4
	0.8	0.42B+0.33B	49.8	82.7	82.9	85.6	33.3	91.3
	0.9	0.42B+0.17B	49.9	82.0	82.6	85.6	33.2	91.3
	0.99	0.42B+0.02B	49.4	80.9	81.6	85.3	32.9	91.4
Med-MoE-Phi (EMNLP 24)	0.0	0.84B+2.52B	55.0	85.3	84.6	85.8	35.1	91.5
	0.2	0.84B+2.68B	55.0	85.3	84.7	85.8	35.0	91.5
	0.4	0.84B+2.01B	55.0	85.3	84.6	86.0	35.1	91.5
	0.6	0.84B+1.34B	55.0	85.3	84.7	86.0	35.1	91.4
	0.8	0.84B+0.67B	55.0	84.6	84.9	85.6	35.2	91.5
	0.9	0.84B+0.34B	55.2	84.6	84.9	85.1	35.0	91.6
	0.99	0.84B+0.03B	55.7	84.9	84.0	85.6	35.0	91.5

Table S10. Detailed results of applying DeRS-Quantization (with different bit width) to compress two vanilla upcycled Med-MoE models on the medical multi-modal task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model. The light-gray Added Params denotes the additional parameters introduced by the universal FFN layers that are not considered as experts of MoE layers.

Vanilla Upcycled MoE Model	Bit Width	Added Params.	VQA-RAD		SLAKE		PathVQA	
			Open	Closed	Open	Closed	Open	Closed
Med-MoE-StableLM (EMNLP 24)	16	0.42B+1.24B	51.0	82.3	82.4	85.3	33.4	91.4
	8	0.42B+0.83B	50.8	82.3	82.3	85.1	33.3	91.4
	4	0.42B+0.42B	50.8	82.3	82.3	85.3	33.3	91.3
	2	0.42B+0.21B	50.5	82.3	82.5	85.3	32.9	91.4
	1	0.42B+0.10B	43.3	80.5	79.5	84.1	31.2	91.1
Med-MoE-Phi (EMNLP 24)	16	0.84B+2.52B	55.0	85.3	84.6	85.8	35.1	91.5
	8	0.84B+1.68B	55.0	85.3	84.6	85.8	35.1	91.5
	4	0.84B+0.84B	54.9	85.3	84.9	86.3	35.1	91.5
	2	0.84B+0.42B	54.6	85.0	84.6	85.6	34.8	91.4
	1	0.84B+0.21B	54.0	83.1	80.2	83.2	31.6	90.7

Table S11. Detailed results of applying DeRS-Sparsification (with different drop rates) to compress the vanilla upcycled Coder-MoE model on the code generation task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model. The light-gray Added Params denotes the additional parameters introduced by the universal FFN layers that are not considered as experts of MoE layers.

Vanilla Upcycled MoE Model	Drop Rate	Added Params.	HumanEval	HumanEval+	MBPP	MBPP+
Coder-MoE (ACL 24)	0.0	0.81B+2.43B	64.6	61.0	63.9	51.4
	0.2	0.81B+2.60B	63.4	60.4	63.7	51.4
	0.4	0.81B+1.95B	63.4	59.8	63.9	51.6
	0.6	0.81B+1.30B	64.0	59.8	64.4	53.1
	0.8	0.81B+0.65B	62.2	59.1	63.7	51.9
	0.9	0.81B+0.32B	62.2	57.3	63.4	51.6
	0.99	0.81B+0.03B	56.7	53.0	56.1	45.6

Table S12. Detailed results of applying DeRS-Quantization (with different bit width) to compress the vanilla upcycled Coder-MoE model on the code generation task. **Added Params** represents the number of additional parameters of the compressed MoE model compared to its corresponding dense model. The light-gray Added Params denotes the additional parameters introduced by the universal FFN layers that are not considered as experts of MoE layers.

Vanilla Upcycled MoE Model	Bit Width	Added Params.	HumanEval	HumanEval+	MBPP	MBPP+
Coder-MoE (ACL 24)	16	0.81B+2.43B	64.6	61.0	63.9	51.4
	8	0.81B+1.62B	64.0	60.4	63.7	51.6
	4	0.81B+0.81B	63.4	59.8	63.7	52.1
	2	0.81B+0.41B	64.0	61.0	62.4	51.1
	1	0.81B+0.20B	9.1	9.1	6.8	6.3

Table S13. Detailed training hyper-parameters and our DeRS upcycling hyper-parameters for experiments on three tasks. **DeRS-SM Rate** denotes the sparse rate for the Sparse-Matrix-based DeRS upcycling while **DeRS-LM Rate** denotes the rank for the Low-rank-Matrix-based DeRS upcycling. † denotes the extended DeRS upcycling implementation.

Config	Task		
	General Multi-Modal	Medical Multi-Modal	Code Generation
Training Epochs	1	9	4
Learning rate	2e-5	2e-5	5e-5
Learning rate schedule	Cosine	Cosine	Linear
Training Batch size per GPU	4	8	4
Gradient Accumulation Steps	4	2	2
Number of GPU	8 × A100 (80G)	4 × A100 (80G)	8 × A100 (80G)
Precision	Bfloat16	Bfloat16	Bfloat16
DeRS-SM Rate	0.9999	0.9999	0.9
DeRS-LM Rank	1	1	4
DeRS-SM† Rate	-	0.999	0.9
DeRS-LM† Rank	-	4	4