2025 Samsung AI Challenge

거대 모델의 성능 저하 없이 크기를 줄이는 방법

팀명	프로메테우스
팀원 성명	김도현, 윤상민, 정연석

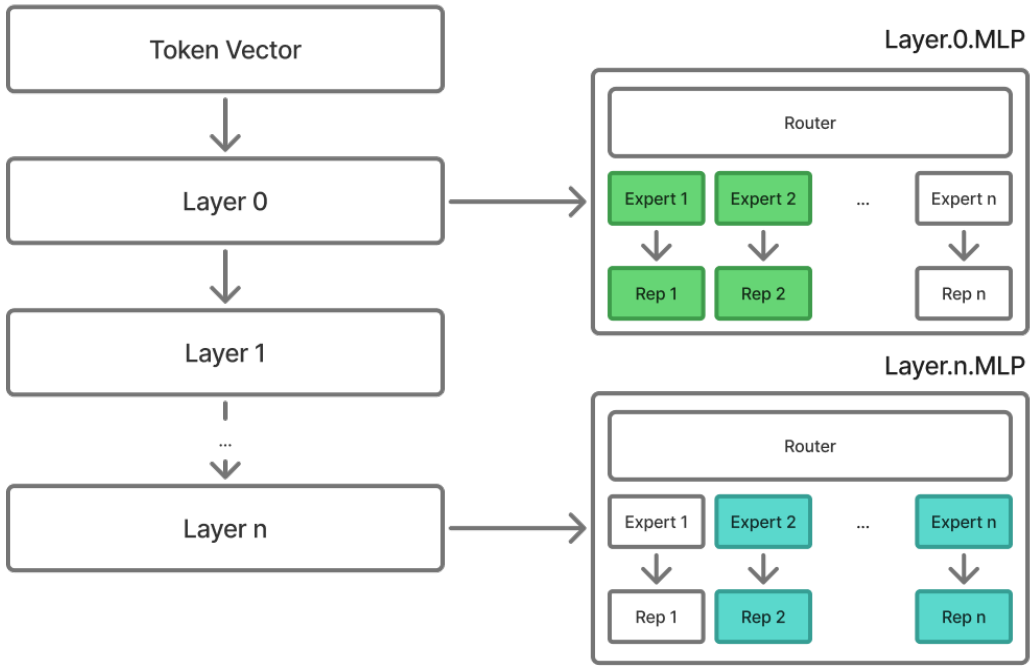
아이디어 기획서

[작성방법]

기획서는 예선 평가 심사 기준에 맞추어 작성
심사 기준 항목 내에서 페이지 구성은 자유

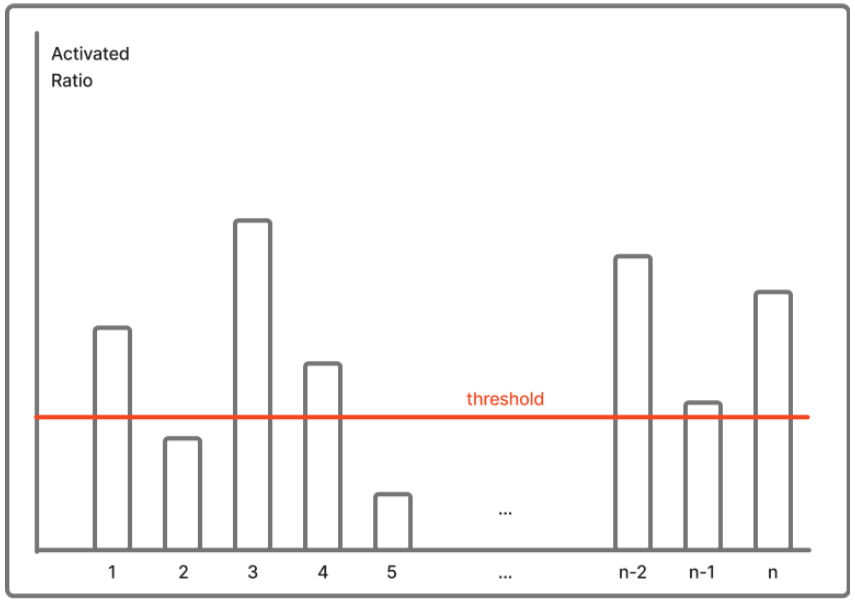
1. 방법론 타당성 및 효용성 1.1. 프레임워크

Step 1



외부 데이터셋을 이용하여 활성화된 Expert의 Representation 얻기

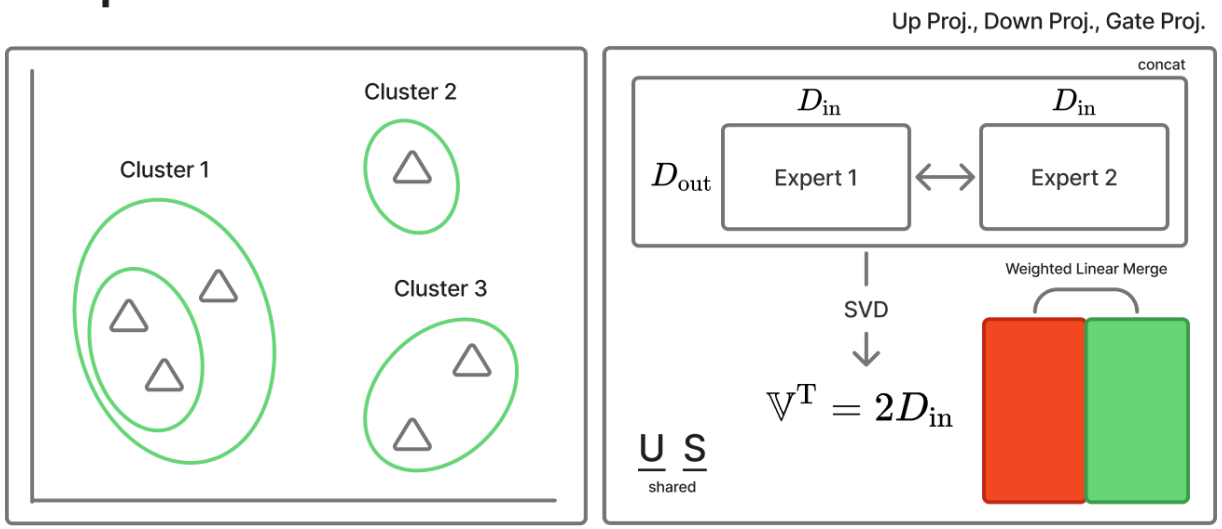
Step 2



Router Logit을 이용한 frequency 기반 Expert Pruning

1. 방법론 타당성 및 효용성 1.1. 프레임워크

Step 3



Step 3: Expert Merging based on Representation Clustering

Step 3.1 clustering

Step 3.2.1 training free

Step 3.2.2 training + soft merging

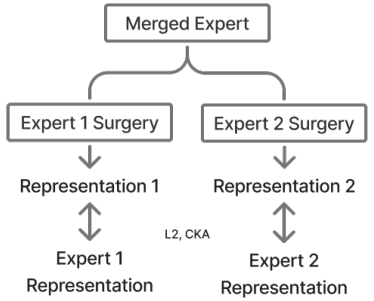
Training

1. Adaptive Merging

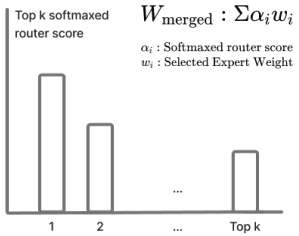
$$\text{Merged } w = \Sigma \frac{\alpha_i w_i}{\Sigma \alpha_i}$$

α_i: Ratio for each Expert
trained by CMA - ES Algorithm

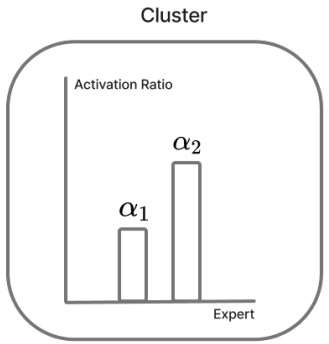
2. Representation Surgery



Soft Merging



Training-Free



$$\text{Merged } w = \Sigma \frac{\alpha_i w_i}{\Sigma \alpha_i} \oplus$$

Adjust # of Activated Experts

α_i: Activation Ratio at each Expert

1. 방법론 타당성 및 효용성 1.2. 외부 데이터

1.2.1. (방법 1) 사전학습 데이터

WikiText-2, PTB, C4 등의 사전학습 데이터에서 무작위 샘플링

- LLM 사전학습 데이터셋은 이미 존재하는 여러 데이터셋 중 가장 특정 도메인에 치우쳐져 있지 않은 데이터셋으로, 도메인 전반의 데이터 분포를 활용하는데 도움이 됨.
- HC-SMoE, Sub-MoE를 제안한 연구에서도 사전학습 데이터를 이용하여 각 Expert의 Representation을 얻으며, 이는 여러 벤치마크에서 검증된 성능을 보여줌.

1.2.2. (방법 2)Token Inference 데이터

LLM 사전학습 데이터의 한계점

- 무작위로 샘플링된 사전학습 데이터셋에도 어느 정도의 도메인이 포함되어 있을 가능성이 있어 어떤 도메인에서는 잘 작동하지 않을 가능성이 있음.
- 무작위로 샘플링된 사전학습 데이터셋에도 어느 정도의 도메인이 포함되어 있을 가능성이 있어 어떤 도메인에서는 잘 작동하지 않을 가능성이 있음.

LLM 스스로 생성한 데이터 이용

- 하나의 토큰을 무작위로 LLM에 넣어 Beam Search 기반으로 확률이 높고 연속된 토큰들의 집합(짧은 길이)을 K개 생성함. 해당 과정을 반복하여 다양한 연속된 토큰들의 집합을 생성하고, 최종적으로 이를 LLM에 넣어 문장 데이터 생성한 후 활용
- 실제로 LLM이 지니고 있는 전반적인 도메인을 활용하면서도 실제 출력 분포와 같은 데이터를 이용하여 해당 LLM 내부 모듈의 병합 단계에서 편향이 감소하고 조금 더 정렬된 효과를 가져올 수 있을 것이라고 생각함.

1. 방법론 타당성 및 효용성 1.3. Expert Pruning

토큰 생성에 크게 기여하지 않는 Expert만 제거

- LLM에 외부 데이터를 넣었을 때 얻을 수 있는 Router Logit을 통해 각 Expert의 활성화되는 빈도를 구하고, 특정 threshold를 이하의 Expert를 제거함.
- 이를 통해, 총 Expert의 수를 줄이면서도 성능 저하를 최소화시킬 것이라고 생각함.

1. 방법론 타당성 및 효용성 1.4. Expert Merging based on Representation Clustering

1.4.1. Expert Representation-based Clustering

Expert Representation 기반 유사도 활용

- Router Score는 특정 입력에 대해 어떤 Expert가 선택되었는지에 대한 정보만을 제공하며, 가중치는 동일한 기능을 수행해도 다른 파라미터 공간에 위치할 수 있음. 이는 실제 Expert가 만들어내는 출력 분포의 의미나 기능적 유사성을 반영하지 못하기 때문에, Router Score나 가중치 유사도를 클러스터링에 활용하는 것은 성능 저하를 초래할 수 있음.
- Expert가 만들어내는 출력은 실제 태스크에서 Expert가 맡고있는 기능을 반영하고 있기 때문에, Expert 병합 과정에서 이를 활용하는 것이 기능적인 측면에서 더 잘 작동할 것이라고 생각함.

오직 Activated Expert Representation만 활용

- 지금까지의 연구에서는, 입력이 들어왔을 때 활성화되지 않은 Expert의 표현까지도 활용하여 클러스터링을 진행함. 하지만, 특정 입력에 대해서 활성화되지 않은 Expert는 학습 때도 활용되지 않았을 가능성이 높기 때문에 상당히 많은 잡음을 가지고 있을 확률이 높고, 이는 클러스터링 단계에서 성능 저하를 일으킬 수 있다고 판단함.
- 따라서, 외부 데이터를 입력으로 넣었을 때 활성화된 Expert의 표현만 활용하여 Expert 클러스터링을 진행하여 필요한 메모리를 줄이면서 더욱더 안정적인 Expert 클러스터를 생성할 예정임.

1. 방법론 타당성 및 효용성 1.4. Expert Merging based on Representation Clustering

1.4.1. Expert Representation-based Clustering

Hierarchical Clustering 선정

- **클러스터 개수 사전 지정 불필요**: K-means와 같은 클러스터링 방식은 클러스터의 개수를 사전이 미리 정해져 있어야 함. 하지만, Layer마다 Expert의 출력 분포 다양성은 다르기 때문에 최적의 클러스터 개수를 찾기는 어려움. 계층적 클러스터링은 특정 threshold에 따라 유동적으로 클러스터가 생성되기 때문에 해당 문제의 영향을 크게 안받음.
- **안정적, 점진적 병합**: 계층적 클러스터링은 coarse에서 fine 순서로의 안정적인 병합 경로를 제공함. 또한, 하이퍼파라미터 초기화 민감도가 낮아 재현성이 높음.
- **다양한 데이터 분포에서의 활용**: Expert의 출력 분포가 비선형적, 비구형적이라도 상대적으로 강건하고, 분포의 형태에 크게 제약 받지 않음.

Distance Metric 선정

- **L2**: 일반적으로 많이 사용되는 메트릭이며, 값의 크기 차이를 잘 포착함. 하지만, 단순 크기 차이에 민감하기 때문에 표현의 구조적, 의미적 유사성은 반영되기 어려움.
- **CKA**: 신경망의 두 표현이 얼마나 유사한지를 측정하는 척도로, 직교 변환, 뉴런 순서, 스케일링에 불변하다는 특징을 가지고 있음. 스케일링에 불변하기 때문에 절대적인 값 크기 차이를 구분하는 데는 약함.

이 두 개의 메트릭을 함께 사용함으로써 표현의 크기 차이, 구조적, 의미적 유사성을 동시에 고려할 수 있음.

1. 방법론 타당성 및 효용성 1.4. Expert Merging based on Representation Clustering

1.4.2. Expert Merging

1.4.2.1. (공통) SVD 기반 Merging 기법

- 같은 클러스터 Expert들의 가중치를 연결한 후 SVD를 수행하여, Expert들이 서로 공유하는 subspace인 U 행렬과 각 Expert별 특성을 지닌 V 행렬들을 생성한 후, 각 Expert별 V 행렬들을 가중합한 후 U 행렬과 재구성하여 하나의 병합된 Expert 가중치를 얻을 수 있음.

→ $U\Sigma[V\_merged]^T$

- 각각 다른 태스크, 도메인에 특화된 Expert들은 서로 다르게 정렬된 표현을 가지고 있을 가능성이 높아, 각 Expert들을 하나의 공유된 공간으로 정렬 후 병합을 통해 성능 저하 최소화를 시도함.

관련 연구

- LoRA Merging 연구 분야에서 낮게 정렬된 표현을 가지는 LoRA 어댑터들을 병합하기 위한 방법으로, SVD를 활용한 subspace 공유를 통해 정렬된 상태에서 병합을 진행하는 방식인 KnOTS
- 이후, 같은 방식의 Expert Merging 방법으로 Sub-MoE가 제안되었음.

1.4.2.2. (방법 1) Training-Free + 활성화되는 Expert 수 조정

- 각 Expert별 V 행렬들을 가중합하는 과정에서, 외부 데이터를 통해 얻은 각 Expert의 활성화 빈도를 가중치로 활용함. 이는 병합 과정에서 자주 활용되고 기여하는 Expert의 특성 및 역할을 강조할 수 있음.
- Expert 병합이 끝난 후, 활성화되는 Expert 수를 조정하고 벤치마크 평가를 반복함으로써 최적의 개수를 구함.

1. 방법론 타당성 및 효용성 1.4. Expert Merging based on Representation Clustering

1.4.2. Expert Merging

1.4.2.3. (방법 2) Training + Soft Merging

Training Objective

- Expert Merging & Pruning 후 최대한 성능을 유지하기 위해서는 그 이전의 표현과 최대한 가깝게 만드는 것이 최종 목표임.
- Expert Merging & Pruning 전 외부 데이터를 통해 얻은 표현들을 정답 레이블로 활용한 추가적인 학습 과정이 있다면, 상기 과정 후 모델의 성능을 최대한 유지하는데 많은 도움을 줄 것으로 보임.
- Expert 표현의 유사도에 대한 기준은 클러스터링에서 이용한 메트릭과 같이 L2와 CKA를 함께 사용함으로써 표현의 크기 차이, 구조적, 의미적 유사성을 동시에 고려하고자 함.

Merging Coefficient 학습

- 각 Expert별 V 행렬들을 가중합하는 과정에서, 각 Expert에 대한 가중치(병합 비율)를 역전파 알고리즘이나 CMA-ES 알고리즘 등을 활용하여 학습시킴. 이는 모델 스스로 세밀하게 조정된 Expert Merging을 가능하게 만들어 성능 저하를 최소화시킬 수 있음.

1. 방법론 타당성 및 효용성 1.4. Expert Merging based on Representation Clustering

1.4.2. Expert Merging

1.4.2.3. (방법 2) Training + Soft Merging

Expert-specific Representation 모듈

- 병합된 Expert 뒤에 기존의 각 Expert를 위한 Lightweight Expert-specific 모듈을 붙여 각각의 Expert가 생성하던 표현과 최대한 가깝게 변형될 수 있도록 하여 기존 모델의 표현력을 유지시킴. 해당 모듈은 역전파 알고리즘으로 학습될 수 있으며, 모델에 추가적인 가중치가 붙긴하지만 Lightweight 모듈이기 때문에 총 파라미터 수는 크게 줄이면서도 성능 저하 또한 최소화시킬 수 있을 것이라고 생각함.

Soft Merging

- 순전파 과정에서 활성화되는 Expert들의 가중치를 Router Score를 기반으로 가중합하여 하나의 Expert로 합치는 방식으로, 활성화되는 Expert들의 지식을 하나의 Expert로 합칠 수 있게 되면서도 실질적으로 활성화된 파라미터의 수는 하나의 Expert가 활성화된 것과 같은 효과를 주게됨.
- 이 과정에서 일어나는 표현력에 관한 손실은 Merging Coefficient와 Expert-specific Representation 모듈 학습을 통하면서 함께 어느정도 극복될 수 있을 것이라고 생각함.

MMLU, GSM8K 등 대표적인 LLM 벤치마크를 활용하여 각각의 방법들을 실험해보고 최적의 방법을 확인하고자 함.
이때, 해당 방법들 중에서 대회 규칙에 위반되는 사항이 있는지 주최측에 확인해보고 가능한 방법들만 시도할 예정임.

2. 구현 가능성 2.1 Representation 추출

- LLM 사전학습 데이터는 Huggingface Dataset에서 오픈 소스로 다운로드 받을 수 있으며, 실제 추론을 통해 Self-Generated 데이터를 얻는 과정은 약간의 알고리즘과 Huggingface Text-Generation Pipeline을 통해 쉽게 구현이 가능함.

- 모델에 입력을 넣었을 때 활성화된 Expert의 Representation을 얻는 과정은 Huggingface에서 불러온 모델에서 내부적으로 forward 함수를 수정하면 가능함.

2. 구현 가능성 2.2 Expert Pruning

- Huggingface에서 불러온 모델에 입력을 넣을 때 각 Layer의 Router Logit 또한 출력하게 하는 인자가 존재함. 이를 활용하면 각 입력에 대한 Router Logit을 구할 수 있고 각 Expert별 활성화되는 빈도 비율을 또한 쉽게 구할 수 있음.

- Expert Pruning을 진행할 때, 제거해야 할 Expert의 index만 구한다면 해당 Expert 전체와 Router의 가중치 일부분만을 제거한 후, 새롭게 mlp 모듈만 초기화하면 되고 기존의 forward 함수도 그대로 사용이 가능하여 크게 문제없음.

2. 구현 가능성 2.3 Expert Merging based on Representation Clustering

2.3.1 Hierarchical Clustering

- Hierarchical Clustering은 각 Expert끼리의 distance 행렬만 구하면은 구현하는데 크게 어렵지 않고, 메트릭으로 L2는 쉽게 구현이 가능하며, CKA 또한 github에 코드가 올라와 있는 것을 확인함.

2.3.2 Expert Merging

- SVD를 이용한 Merging 방식은 torch 라이브러리를 이용하면 쉽게 구현 가능함. 병합된 Expert를 실제로 모델에 적용하는 것은 Router 가중치는 그대로 두되, 기존에 선택된 Expert가 새롭게 바뀐 Expert 구조에 매핑될 수 있게하는 알고리즘을 forward 함수에 구현하면 크게 문제될 것 같지 않음.
- Merging Coefficient 또는 Expert-specific Representation 모듈을 학습시키는 것과 관련해서는 추가적인 파라미터들을 관련된 모듈에 추가하고 forward 함수를 수정하는 방식으로 가능함. Expert Pruning & Merging 전후 최종 representation을 각각 예측값, 정답 레이블값으로 설정하고 L2와 CKA를 활용한 손실 함수만 정의한다면 학습 또한 문제 없을 것 같음. 역전파 알고리즘은 torch 라이브러리를 이용하면 되고, CMA-ES 알고리즘은 github에 코드가 올라와 있는 것을 확인함.
- Soft Merging 또한 관련 모듈의 forward 함수를 수정하는 방식으로 쉽게 구현이 가능함.

3. 확장 가능성 3.1 MoE 계열 모델에서의 전반적인 사용 가능성

3.1.1 Expert 유형 및 구조와 무관

- Expert Merging을 위한 클러스터링에서 각 Expert의 출력만을 사용함. 이는 Expert의 구조와 관계 없이 어디에서나 활용될 수 있음을 의미함.
- SVD 기반 Merging 기법은 Expert 간 매핑되는 가중치 쌍끼리 활용됨. 이는 Expert의 구조가 어떻든 같은 모델 내의 Expert끼리 가중치 형태만 동일하면 활용 가능하다는 것을 의미함. Merging Coefficient 또한 동일하게 적용됨.
- Expert-specific Representation 모듈은 adapter처럼 단순히 병합된 Expert 뒤에 붙는 모듈임. 따라서, Expert 구조가 어떻든 출력만 얻을 수 있다면 활용가능함.

3.1.2 활성화되는 Expert의 수가 정해진 Routing 전략에서 전반적으로 사용가능

- Expert Pruning & Merging에서 외부 데이터를 활용하여 활성화되는 Expert의 representation과 각 Expert의 활성화 빈도수를 이용함. 활성화되는 Expert의 수가 정해져만 있다면 문제 없이 활용가능함.

- Router가 입력에 대한 Logit과 topk index를 출력하고, Layer별 동일한 입출력 차원을 가지는 Expert들의 집합이 있는 구조라면 해당 방법을 사용할 수 있고, 대부분의 MoE 계열의 모델들이 이에 해당함.
- 따라서, 제안하는 프레임워크는 특정 MoE 변형에 묶이지 않고, “expert representation → pruning → clustering/merging” 이라는 파이프라인으로 일반화할 수 있음.

Reference

1. Muqeeth, M., Liu, H., & Raffel, C. (2023). Soft merging of experts with adaptive routing. arXiv preprint arXiv:2306.03745.
2. Li, L., Qiyuan, Z., Wang, J., Li, W., Gu, H., Han, S., & Guo, Y. (2025). Sub-MoE: Efficient Mixture-of-Expert LLMs Compression via Subspace Expert Merging. arXiv preprint arXiv:2506.23266.
3. Yang, E., Shen, L., Wang, Z., Guo, G., Chen, X., Wang, X., & Tao, D. (2024). Representation surgery for multi-task model merging. arXiv preprint arXiv:2402.02705.
4. Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23(120), 1-39.
5. Chen, I., Liu, H. S., Sun, W. F., Chao, C. H., Hsu, Y. C., & Lee, C. Y. (2024). Retraining-Free Merging of Sparse MoE via Hierarchical Clustering. arXiv preprint arXiv:2410.08589.
6. Liu, E., Zhu, J., Lin, Z., Ning, X., Blaschko, M. B., Yan, S., ... & Wang, Y. (2024). Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs. arXiv preprint arXiv:2407.00945.
7. Li, P., Zhang, Z., Yadav, P., Sung, Y. L., Cheng, Y., Bansal, M., & Chen, T. (2023). Merge, then compress: Demystify efficient smoe with hints from its routing policy. arXiv preprint arXiv:2310.01334.
8. Zhao, H., Qiu, Z., Wu, H., Wang, Z., He, Z., & Fu, J. (2024). Hypermoe: Towards better mixture of experts via transferring among experts. arXiv preprint arXiv:2402.12656.

Reference

8. Xie, Y., Zhang, Z., Zhou, D., Xie, C., Song, Z., Liu, X., ... & Xu, A. (2024). Moe-pruner: Pruning mixture-of-experts large language model using the hints from its router. arXiv preprint arXiv:2410.12013.
9. Lu, X., Liu, Q., Xu, Y., Zhou, A., Huang, S., Zhang, B., ... & Li, H. (2024). Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. arXiv preprint arXiv:2402.14800.
10. Nguyen, D. V., Nguyen, M. H., Nguyen, L. Q., Teo, R. S., Nguyen, T. M., & Tran, L. D. (2025). Camex: Curvature-aware merging of experts. arXiv preprint arXiv:2502.18821.
11. Stoica, G., Ramesh, P., Ecsedi, B., Choshen, L., & Hoffman, J. (2024). Model merging with svd to tie the knots. arXiv preprint arXiv:2410.19735.
12. Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., & Tao, D. (2023). Adamerging: Adaptive model merging for multi-task learning. arXiv preprint arXiv:2310.02575.