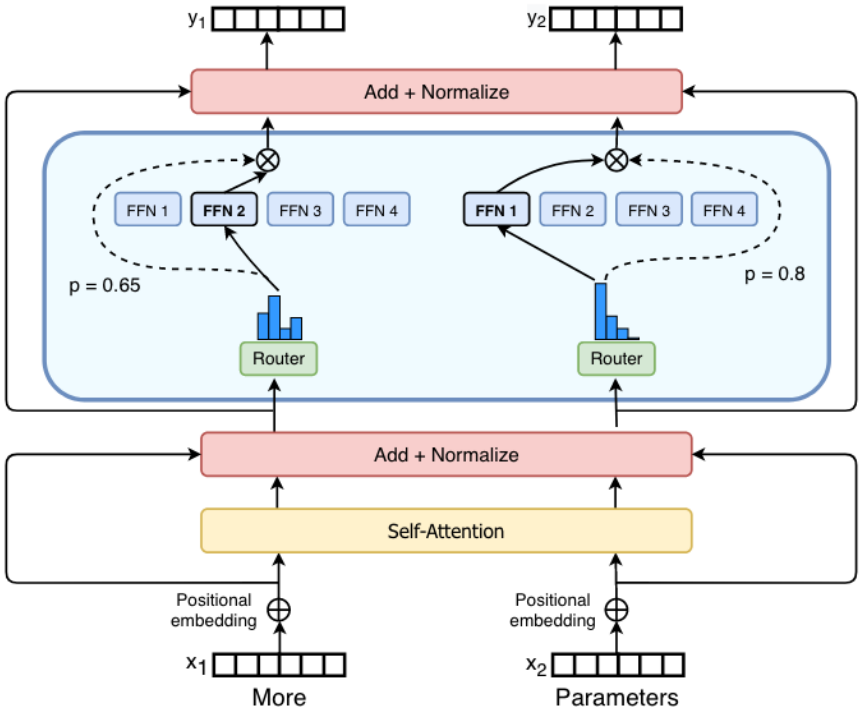
ReMoE: Fully Differentiable Mixture-of-Experts with ReLU Routing

Ziteng Wang, Jianfei Chen, Jun Zhu

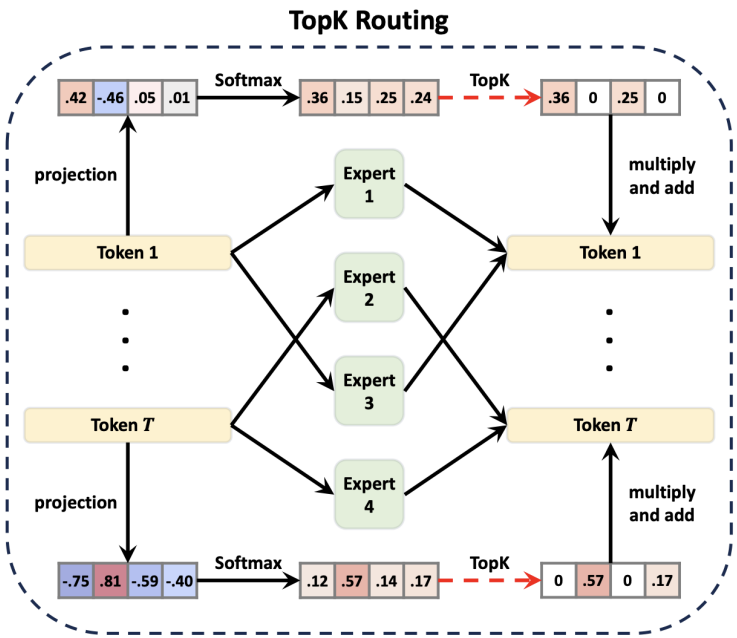
2024.10.24

Mixture-of-Experts in Transformers



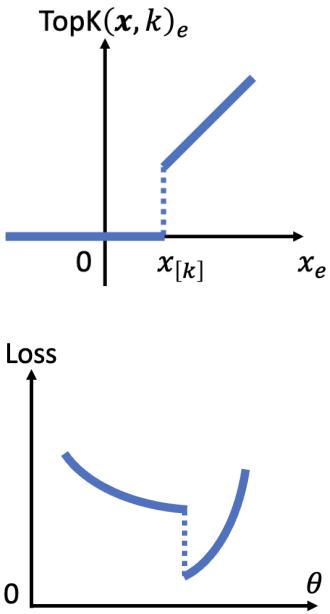
- $\boldsymbol{y} = \sum_{e=1}^{E} R(\boldsymbol{x})_e \mathrm{FFN}_e(\boldsymbol{x})$
- Computation can be saved when $R(\boldsymbol{x})_e = 0$.
- How to calculate $R(\boldsymbol{x})$ is critical.

Typical Routing Method: TopK Routing



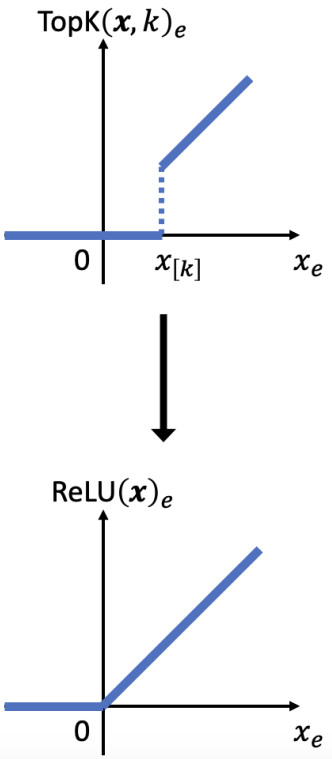
- $R(\boldsymbol{x}) = \text{TopK}(\text{Softmax}(\boldsymbol{x}\boldsymbol{W}), k)$
- Allocate same computation to all tokens. (‘a’ v.s. ‘important’)
- Discontinuous, non-differentiable.
  - $(0.51,\ 0.49) \xrightarrow{\text{Top-1}} (0.51,\ \ 0\ )$
  - $(0.49,\ 0.51) \xrightarrow{\text{Top-1}} (\ \ 0\ ,\ 0.51)$

Discontinuity in TopK



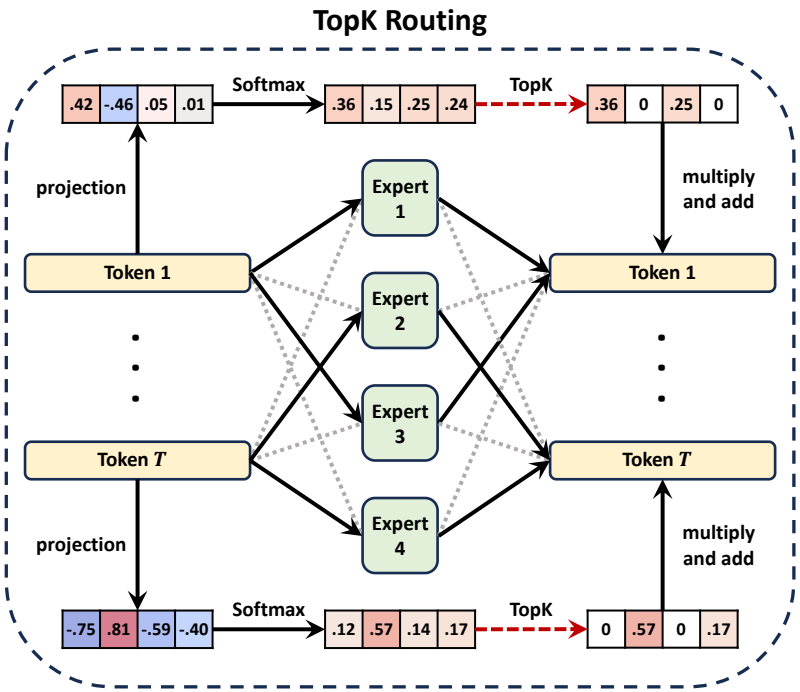
- Let's write TopK function element-wise:
- $$\mathrm{TopK}(\boldsymbol{x}, k)_e = x_e \cdot \mathbf{1}\{x_e \geq t(\boldsymbol{x}, k)\}$$
  $$t(\boldsymbol{x}, k) = \boldsymbol{x}_{[k]}, \text{the } k\text{-th largest element of } \boldsymbol{x}$$
- TopK creates a jump discontinuity at $t(\boldsymbol{x}, k)$.

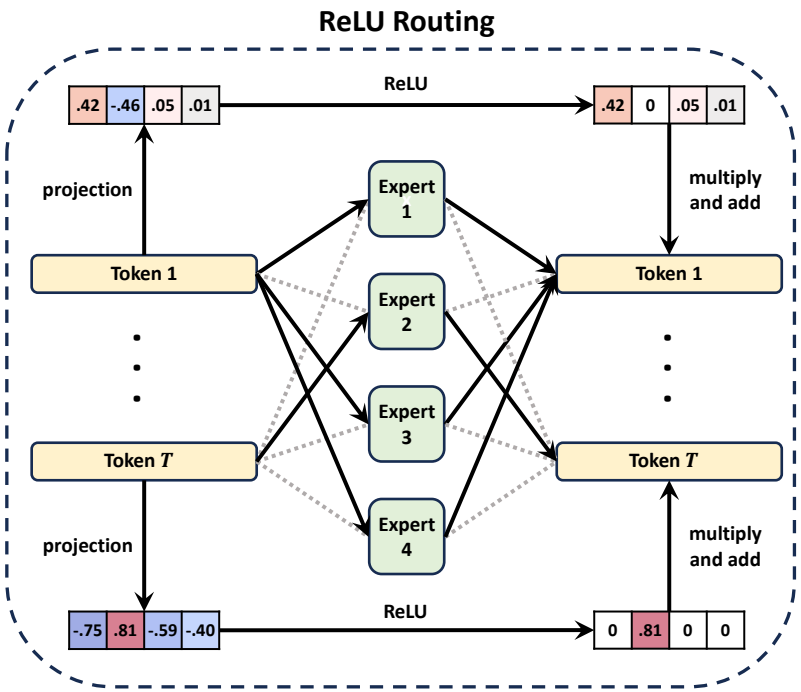
From TopK to ReLU



- If we set $t(\boldsymbol{x}, k) \equiv 0$, TopK becomes ReLU.
  - The breakpoints of all tokens $\boldsymbol{x}$ are aligned.
  - The aligned breakpoints are set to 0, bridging the gap to become continuous and differentiable.
- $\text{TopK}(\boldsymbol{x}, k)_e = x_e \cdot \mathbf{1}\{x_e \geq t(\boldsymbol{x}, k)\}$
- $\text{ReLU}(\boldsymbol{x})_e = x_e \cdot \mathbf{1}\{x_e \geq 0\}$
- This ensures continuous active-inactive transition of experts at $y = 0$.
  - (0.01, -0.01) $\xrightarrow{\text{ReLU}}$ (0.01, 0)
  - (-0.01, 0.01) $\xrightarrow{\text{ReLU}}$ (0 , 0.01)

Comparison of TopK and ReLU



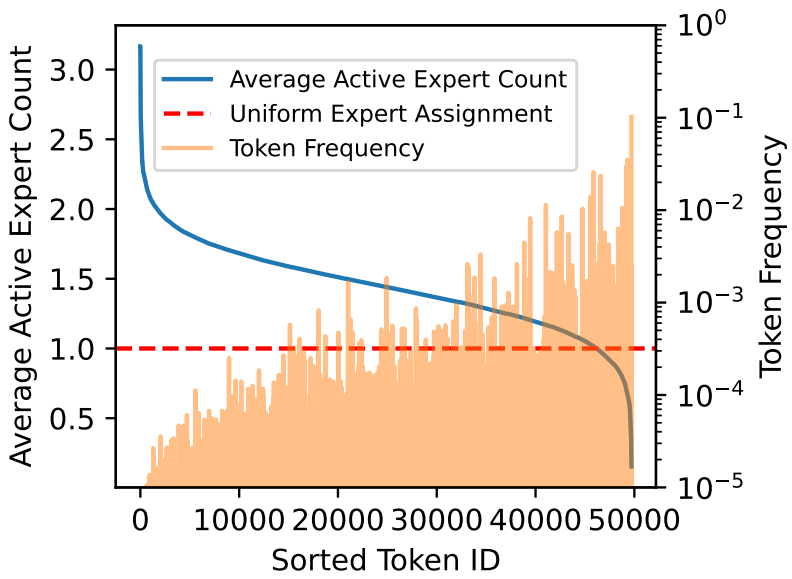


$$R(\boldsymbol{x}) = \text{TopK}(\text{Softmax}(\boldsymbol{x}\boldsymbol{W}), k) \quad \rightarrow \quad R(\boldsymbol{x}) = \text{ReLU}(\boldsymbol{x}\boldsymbol{W})$$

Advantages of ReLU:

- continuous and fully differentiable
- dymamic expert allocation

Dynamic Expert Allocation in ReMoE



- We find the expert allocation of ReMoE strongly related with token frequency.
- A Huffman tree strategy is learned without explicit supervision:
  - "cluster on" common tokens: ' ', 'a', 'the'
  - "encode" rare tokens with richer linear combinations: '©', 'OTAL', '@#'

Controlling Sparsity via Adaptive L_1 Regularization

- ReLU can produce arbitrary zeros. To maintain consistent computation with TopK, we need to control the sparsity to $(1 - \frac{k}{E})$.
- We adopt L_1 regularization on the routing output $R(\boldsymbol{x})$.
- An coefficeint λ_i is adaptively learned with iteration i.
- Total loss: $\mathcal{L} = \mathcal{L}_{lm} + \lambda_i \mathcal{L}_{reg}$
- L_1 regularization loss:

$$\mathcal{L}_{reg} = \frac{1}{LT} \sum_{l=1}^{L} \sum_{t=1}^{T} \left\| R(\boldsymbol{x}_t^l) \right\|_1 = \frac{1}{LT} \sum_{l=1}^{L} \sum_{t=1}^{T} \sum_{e=1}^{E} R(\boldsymbol{x}_t^l)_e$$

- Regularization effect: add $\frac{\lambda_i}{LT}$ to the gradient of each non-zero element of $R(\boldsymbol{x})$, driving it to zero.

Controlling Sparsity via Adaptive L_1 Regularization

- The coefficient λ_i is updated with a zeroth-order algorithm:

$$\lambda_{i+1} = \lambda_i \cdot \alpha^{\text{sign}((1-\frac{k}{E})-S_i)}$$

- where S_i is the sparsity of the current iteration, averaged over all layers and tokens:

$$S_i = 1 - \frac{1}{LTE} \sum_{l=1}^{L} \sum_{t=1}^{T} \sum_{e=1}^{E} \mathbf{1}\{R(\boldsymbol{x}_t^l)_e > 0\}$$

- We set $\lambda_0 = 1e^{-8}$ and $\alpha = 1.2$. But the algorithm is robust to different choices.

λ_0	$1e^{-16}$	$1e^{-12}$	$1e^{-8}$	$1e^{-4}$	1
Valid Loss	2.031	2.029	2.032	2.036	2.032
Settling time	138	136	110	55	92†

† Overshoot observed in 8-92 steps.

α	1.05	1.1	1.2	1.3	1.5
Valid Loss	2.033	2.028	2.032	2.029	2.057*
Settling time	414	211	110	80	52

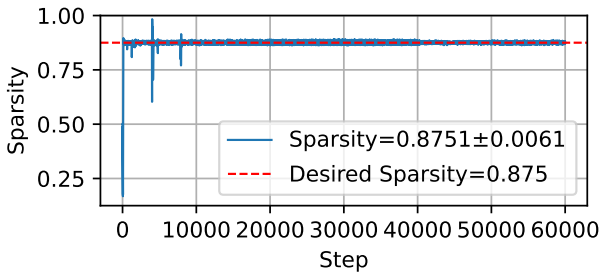
* A large oscillation amplitude in sparsity is observed.

Integrate Load Balancing into L_1 regularization

- Load balancing: experts recieve equal number of tokens.
- Load balancing refined L_1 regularization:

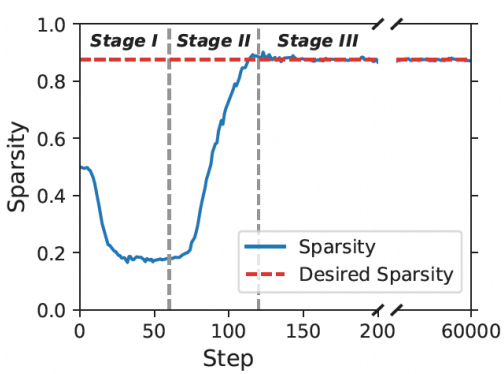
$$\mathcal{L}_{reg,lb} = \frac{1}{LT} \sum_{l=1}^{L} \sum_{t=1}^{T} \sum_{e=1}^{E} f_{l,e} R(\boldsymbol{x}_t^l)_e$$

$$f_{l,e} = \frac{E}{kT} \sum_{t=1}^{T} \mathbf{1}\{R(\boldsymbol{x}_t^l)_e > 0\}$$

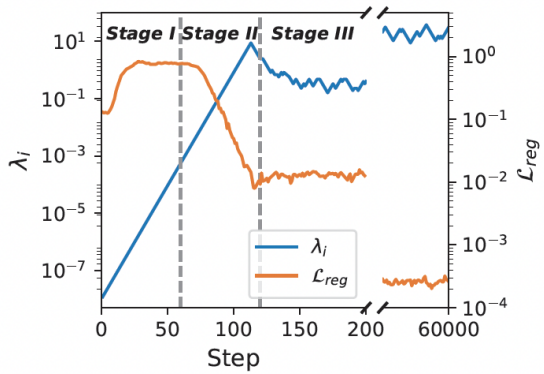


- Notably, this formulation is *identical* to the load balancing loss in TopK routing.

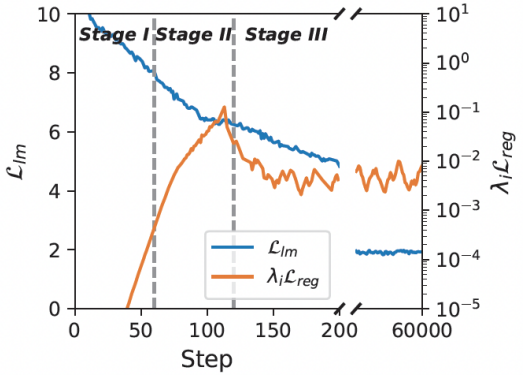
Natural Three-Stage Training in ReMoE



(a) Sparsity S_i



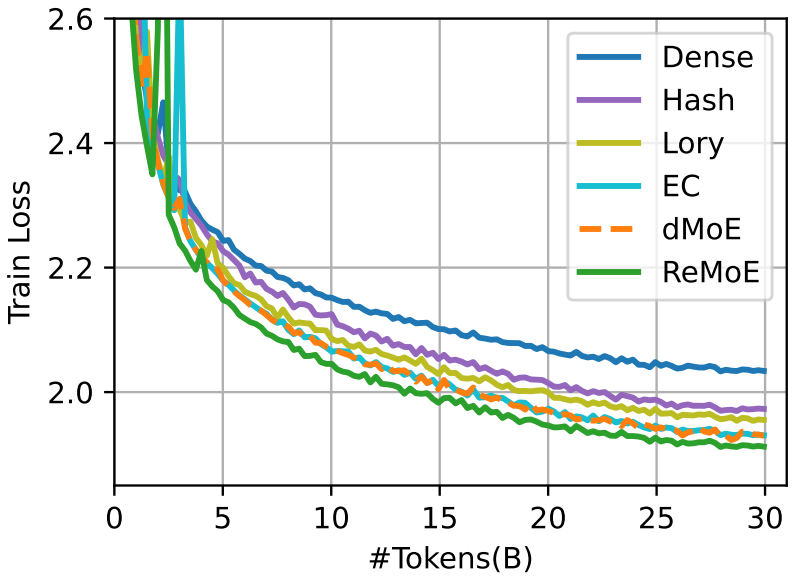
(b) Coefficient term λ_i and regularization term $\mathcal{L}_{reg}$

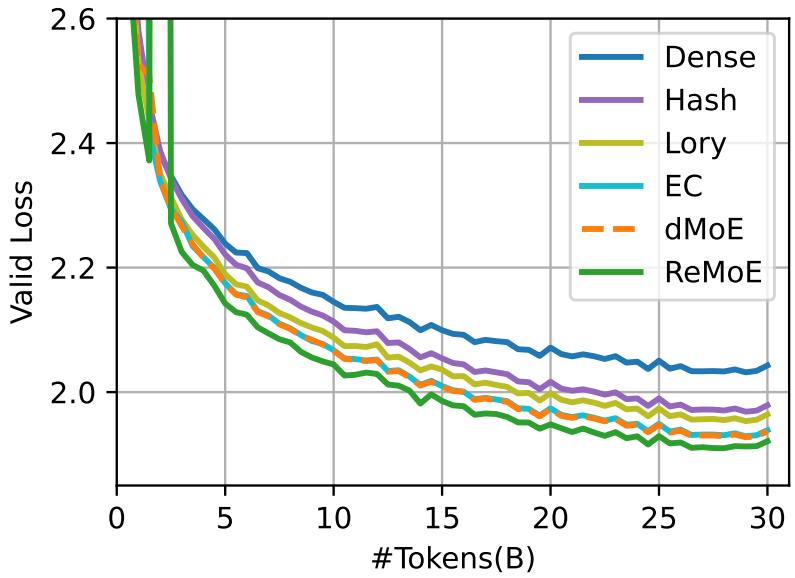


(c) Language model loss $\mathcal{L}_{lm}$ and overall regularization $\lambda_i\mathcal{L}_{reg}$

- Stage I: warm-up stage (dense stage)
- Stage II sparsifying stage (dense to sparse stage)
- Stage III: stable stage (sparse stage)

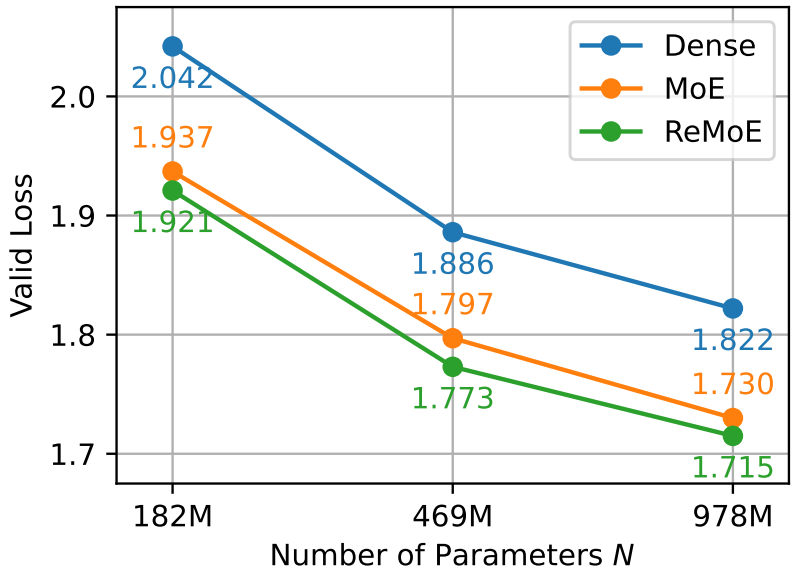
Experiments: Comparison with Other Routing Methods





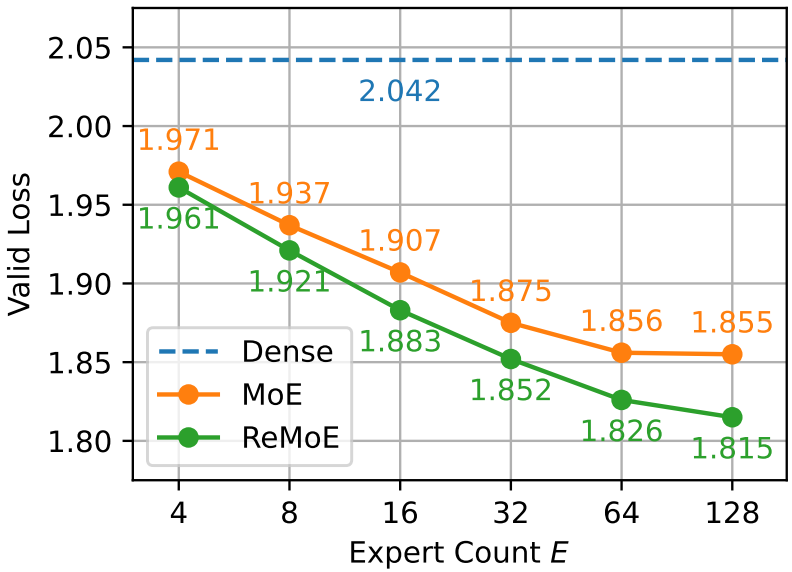
Model	ARC-c	ARC-e	BoolQ	HellaSwag	LAMBADA	PIQA	RACE	Avg.
Dense	19.45	43.35	54.40	28.61	31.09	61.97	28.52	38.20
Hash	19.28	45.45	54.95	29.68	31.44	63.06	27.66	38.79
Lory	**20.31**	42.97	49.54	28.75	32.35	62.24	27.75	37.70
EC	18.86	42.97	**60.21**	29.14	29.26	61.92	27.37	38.53
dMoE	20.05	45.16	57.83	29.83	32.97	**63.55**	28.33	39.67
ReMoE	20.22	**46.68**	54.16	**30.26**	**35.94**	**63.55**	**29.38**	**40.03**

Experiments: Scaling in parameters N



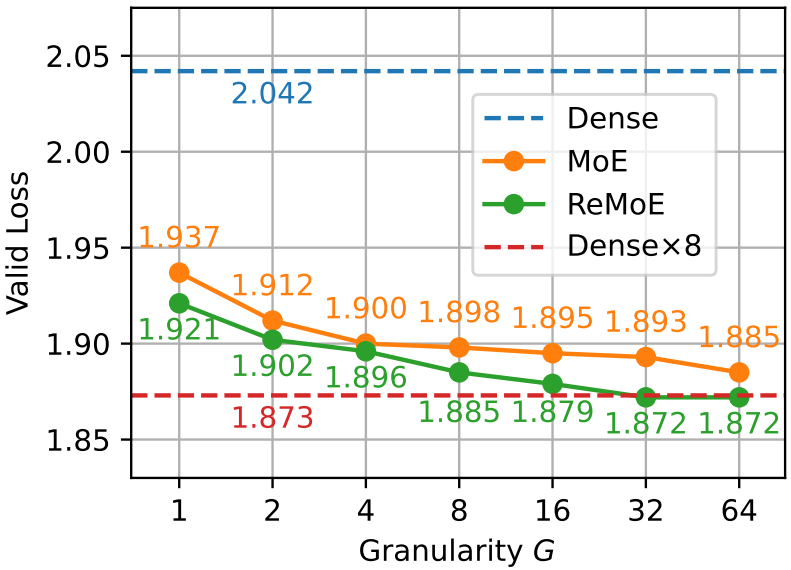
Size	#Parameters	hidden_size	num_layers	num_heads	num_groups	GFLOPs
Small	182M	768	12	12	4	995
Medium	469M	1024	24	16	4	2873
Large	978M	1536	24	16	4	5991

Experiments: Scaling in expert count E



- ReMoE is consistently better.
- ReMoE scales more effectively with E, exhibiting a steeper slope.

Experiments: Scaling in granularity G



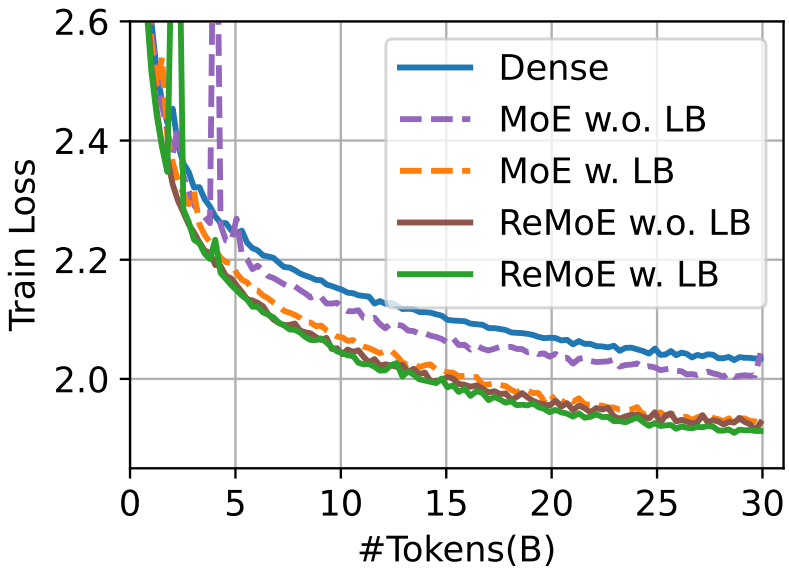
- Granularity G: divide each expert FFN into G smaller experts.

$$\boldsymbol{y} = \sum_{e=1}^{EG} R(\boldsymbol{x})_e \mathrm{FFN}_e(\boldsymbol{x}; d_{ffn}/G)$$

$$R(\boldsymbol{x}) = \mathrm{TopK}(\mathrm{Softmax}(\boldsymbol{x}\boldsymbol{W}), kG)$$

- Fine-grained ReMoE can achieve the theoretical upper bound of Dense×8.

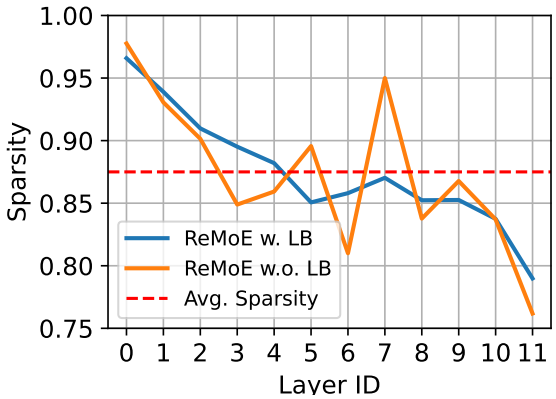
Observations: Role of Load Balancing in ReMoE



- Load balancing (LB) is crucial for MoE.
- But in ReMoE, it is much less important for performance.

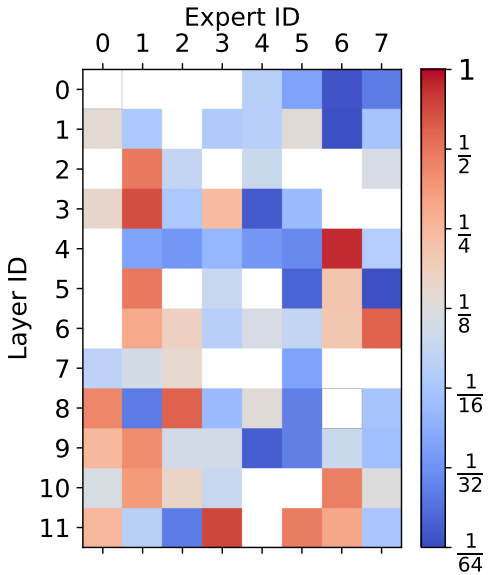
Model	LB	ARC-c	ARC-e	BoolQ	HellaSwag	LAMBADA	PIQA	RACE	Avg.
Dense	-	19.45	43.35	54.40	28.61	31.09	61.97	28.52	38.20
MoE	×	19.20	44.74	50.80	28.60	30.18	62.24	27.94	37.67
MoE	✓	20.05	45.16	**57.83**	29.83	32.97	**63.55**	28.33	39.67
ReMoE	×	19.45	46.34	56.94	30.19	31.79	63.33	28.61	39.52
ReMoE	✓	**20.22**	**46.68**	54.16	**30.26**	**35.94**	**63.55**	**29.38**	**40.03**

Observations: Role of Load Balancing in ReMoE

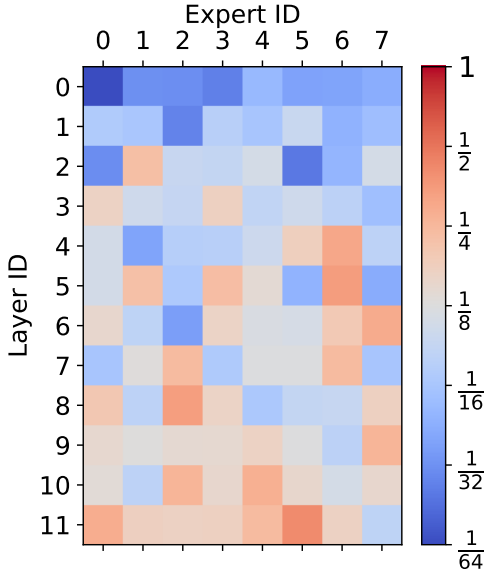


- LB in ReMoE smooths sparsity across layers.
- LB in ReMoE avoids inactive experts.
- More balanced variants with $f_{l,e}^2$, device-level LB, etc. (hardware issue)

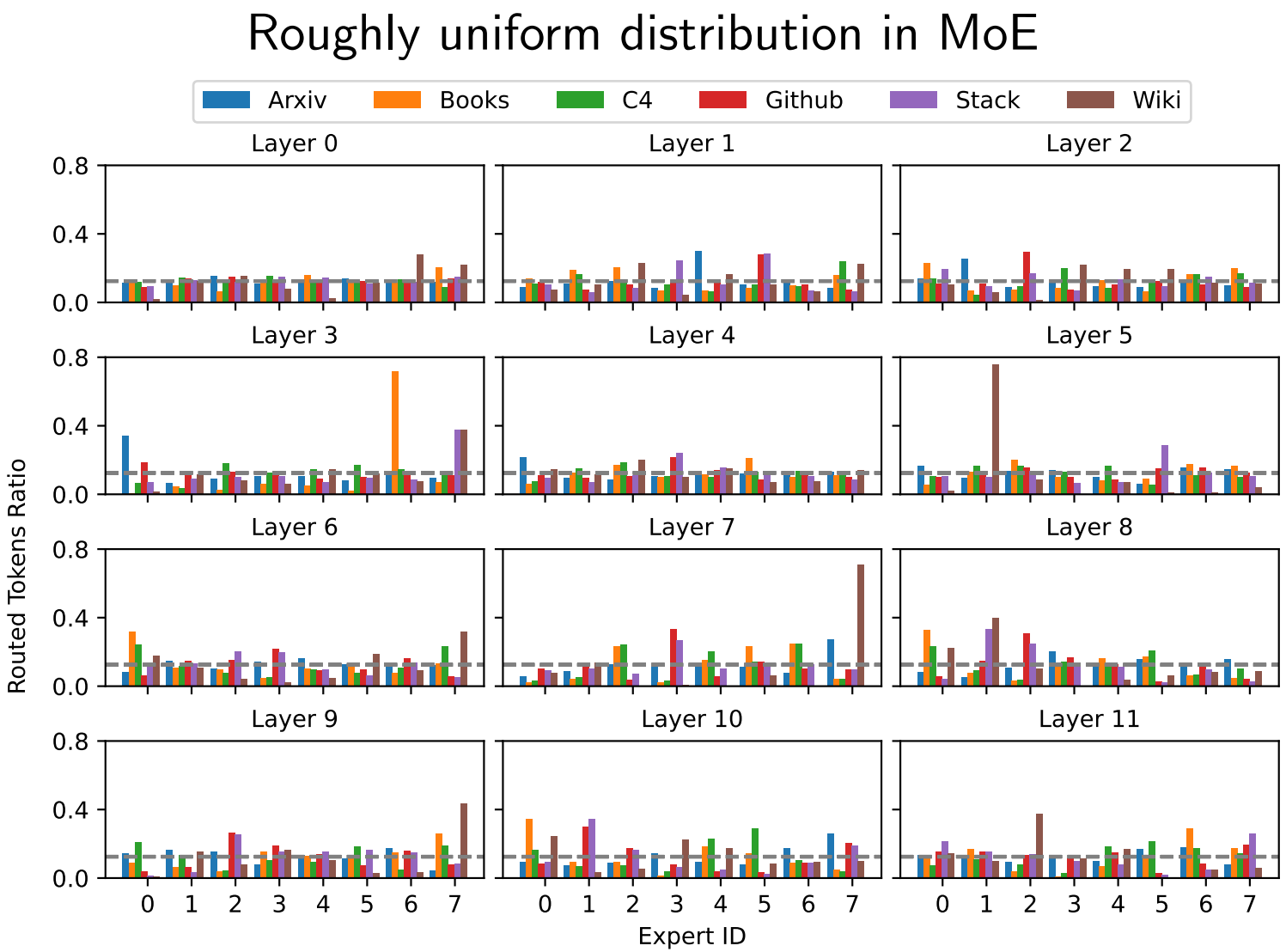
Avg. routed tokens ratio w.o. LB:



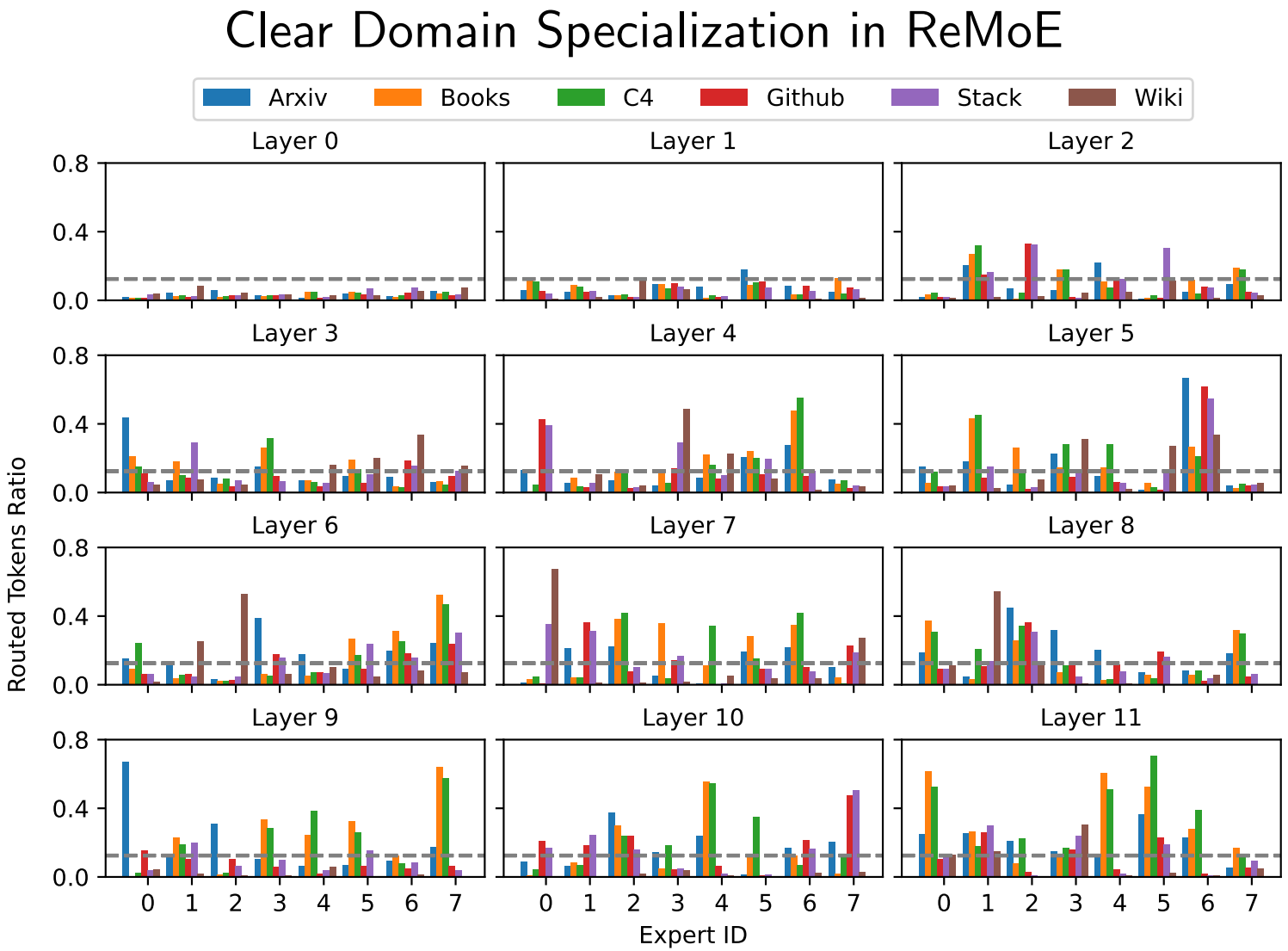
Avg. routed tokens ratio w. LB:



Observations: Domain Specialization in ReMoE



Observations: Domain Specialization in ReMoE



Conclusion

- We propose continuous and differentiable ReLU routing as a drop-in replacement for TopK routing in MoE.
- We introduce adaptive load balancing L_1 regularization and successfully control sparsity.
- ReMoE consistently outperforms TopK-routed MoE across various N, E, and G.
- ReMoE scales better than MoE as E increases.
- The dynamic expert allocation and domain specialization in ReMoE highlight its effectiveness.

Thanks!

Model	N	ARC-c	ARC-e	BoolQ	HellaSwag	LAMBADA	PIQA	RACE	Avg.
Dense	182M	19.45	43.35	54.40	28.61	31.09	61.97	28.52	38.20
	469M	21.50	49.12	56.88	31.12	36.74	64.47	30.53	41.48
	978M	21.93	50.88	60.24	32.42	41.06	67.46	**31.77**	43.68
MoE	182M	20.05	45.16	57.83	29.83	32.97	63.55	28.33	39.67
	469M	22.61	50.63	**60.40**	32.95	39.82	66.27	29.95	43.23
	978M	23.72	53.07	59.42	35.15	43.99	67.63	31.48	44.92
ReMoE	182M	20.22	46.68	54.16	30.26	35.94	63.55	29.38	40.03
	469M	21.67	53.16	58.75	33.80	40.66	67.95	31.20	43.88
	978M	**24.06**	**55.26**	57.28	**35.93**	**44.42**	**68.99**	30.43	**45.20**

Model	E	ARC-c	ARC-e	BoolQ	HellaSwag	LAMBADA	PIQA	RACE	Avg.
Dense	-	19.45	43.35	54.40	28.61	31.09	61.97	28.52	38.20
MoE	4	**21.25**	44.15	**60.06**	29.16	31.69	62.89	28.71	39.70
	8	20.05	45.16	57.83	29.83	32.97	63.55	28.33	39.67
	16	20.82	45.58	44.46	30.56	33.42	64.64	28.42	38.27
	32	19.37	47.18	50.76	31.04	36.02	64.53	28.52	39.63
	64	20.39	48.95	58.96	31.45	36.37	65.51	28.23	41.41
	128	20.99	46.93	57.58	31.90	35.96	65.29	27.56	40.89
ReMoE	4	19.88	46.46	57.43	29.64	33.57	62.95	27.66	39.66
	8	20.22	46.68	54.16	30.26	35.94	63.55	29.38	40.03
	16	20.90	49.28	53.36	30.85	37.09	65.83	**30.05**	41.05
	32	20.56	48.11	59.54	31.42	37.84	65.18	28.42	41.58
	64	20.82	50.51	57.80	32.17	36.74	65.78	27.46	41.61
	128	19.97	**51.05**	56.97	**32.40**	**37.92**	**66.70**	29.86	**42.12**

Model	G	ARC-c	ARC-e	BoolQ	HellaSwag	LAMBADA	PIQA	RACE	Avg.
Dense	-	19.45	43.35	54.40	28.61	31.09	61.97	28.52	38.20
Dense×8	-	**22.78**	48.11	59.66	31.11	35.65	65.02	29.57	**41.70**
MoE	1	20.05	45.16	57.83	29.83	32.97	63.55	28.33	39.67
	2	21.33	46.89	54.74	30.06	32.72	64.20	28.71	39.81
	4	20.48	46.34	54.86	30.56	35.46	64.36	28.61	40.10
	8	21.16	47.14	59.69	30.61	36.77	65.23	26.99	41.08
	16	19.62	48.82	56.54	30.66	35.90	64.74	28.80	40.73
	32	21.08	48.82	58.29	31.18	37.34	64.74	28.04	41.36
	64	20.56	48.15	**60.98**	31.01	**37.40**	64.25	28.13	41.50
ReMoE	1	20.22	46.68	54.16	30.26	35.94	63.55	29.38	40.03
	2	20.14	47.39	57.95	30.60	34.52	63.71	28.52	40.40
	4	20.39	47.94	55.35	31.04	36.11	64.64	29.00	40.64
	8	20.82	48.36	60.49	30.90	36.06	63.87	28.90	41.34
	16	21.25	**49.41**	56.06	30.91	36.23	64.91	29.95	41.25
	32	20.90	48.86	55.81	31.14	36.58	64.69	**30.05**	41.15
	64	20.65	48.74	60.06	**31.56**	36.43	**65.40**	29.00	41.69