NEURAL INFORMATION PROCESSING SYSTEMS

Project page

DictLLM: Post-training Compression of Large Language Model with Dictionary Kernels

Jinho Park[1], Se Young Chun[2], Mingoo Seok[1]
[1]Columbia University
[2]Seoul National University

jp4327@columbia.edu, sychun@snu.ac.kr, ms4415@columbia.edu

COLUMBIA UNIVERSITY

VLSILab @COLUMBIA UNIV

VERI LUX MEA

Table of contents

- Motivation for layer sharing with dictionary
- Proposed method
  - Dictionary-based compression
  - Dictionary-based forward operation
- Experimental Results
  - Perplexity-model size trade-off
  - Benchmark on language modeling and reasoning tasks
  - Ablation studies on pruning target & dictionary update
- Conclusion



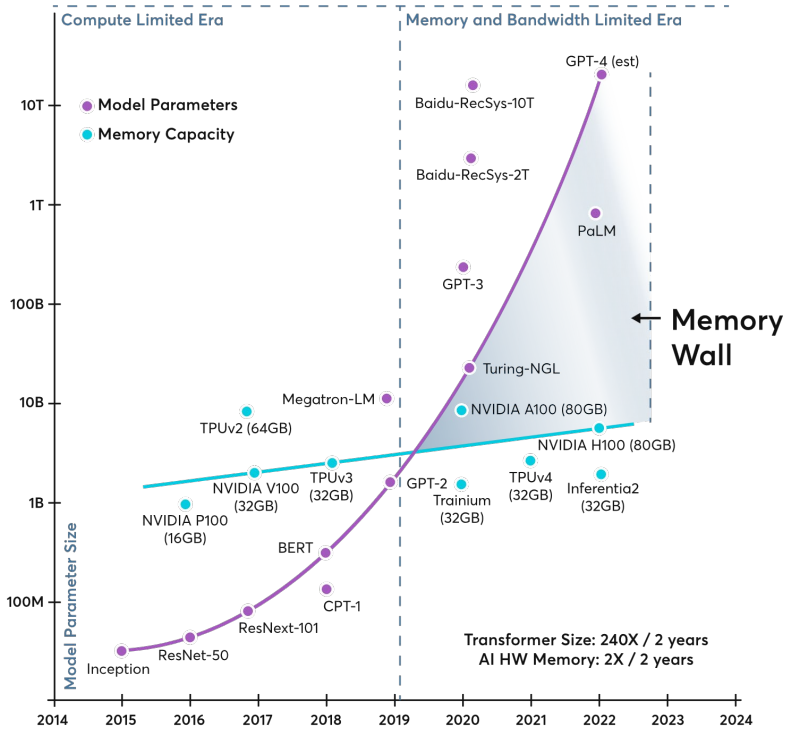
Table of contents

- **Motivation for layer sharing with dictionary**
- Proposed method
  - Dictionary-based compression
  - Dictionary-based forward operation
- Experimental Results
  - Perplexity-model size trade-off
  - Benchmark on language modeling and reasoning tasks
  - Ablation studies on pruning target & dictionary update
- Conclusion

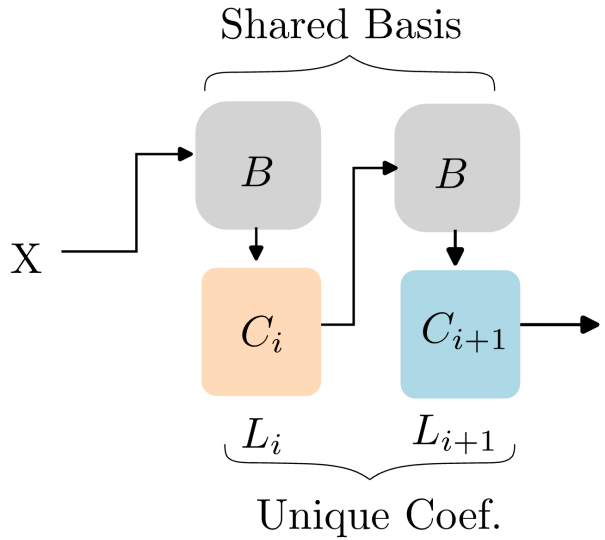


Existing post-training compression method

- **Challenge:** Billion-parameter LLMs impose substantial memory and compute demands on resource-constrained devices.
- **Existing solutions:** SVD-LLM and Basis Sharing reduce model size.



Memory Wall - AyarLabs

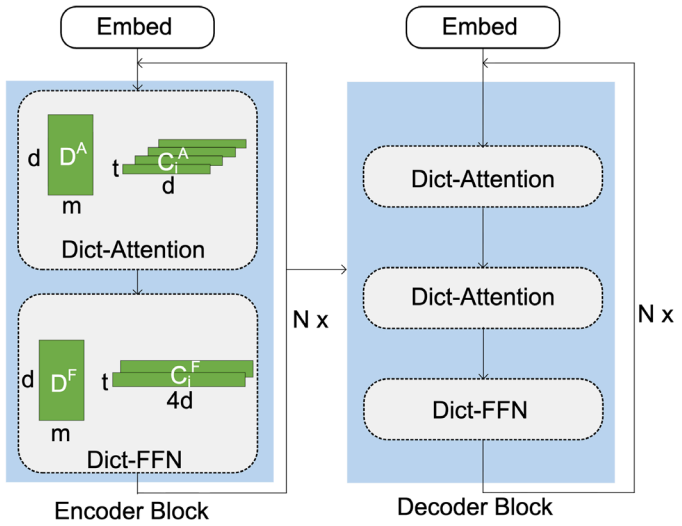


Basis Sharing [ICLR25]

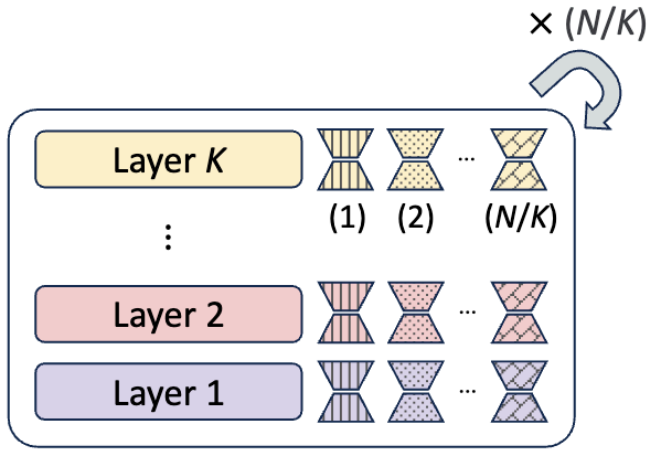
VLSILab
@COLUMBIA UNIV

Existing post-training compression method

- **Research gap:** Dictionary-based weight sharing exploits cross-layer redundancy, but often requires expensive training or fine-tuning.
- **Our approach:** DictLLM enables post-training dictionary compression through shared dictionaries and sparse coefficients, *without* fine-tuning.



DictFormer [ICLR22]



Relaxed recursive transformer [ICLR25]

Table of contents

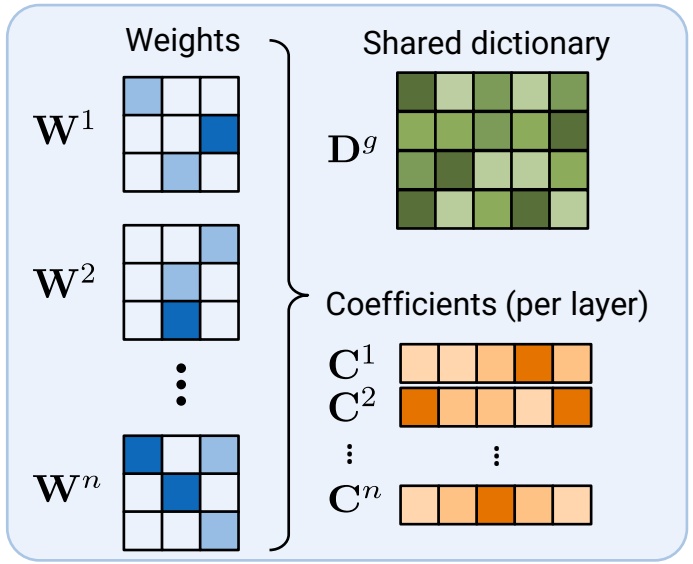
- Motivation for layer sharing with dictionary
- **Proposed method**
  - **Dictionary-based compression**
  - **Dictionary-based forward operation**
- Experimental Results
  - Perplexity-model size trade-off
  - Benchmark on language modeling and reasoning tasks
  - Ablation studies on pruning target & dictionary update
- Conclusion



Dictionary-based compression method

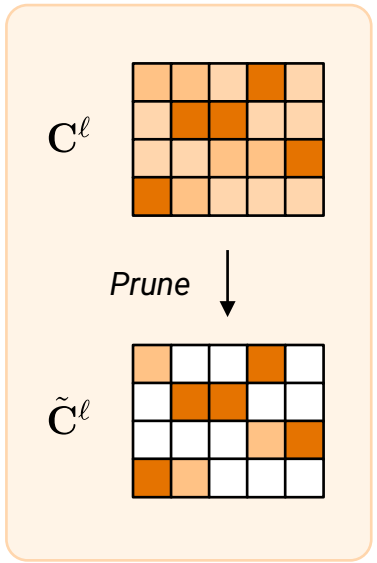
- We propose dictionary representation where we map the network into layer-shared dictionary + layer-specific coefficients.
- The compression follows 3+1 steps.

1 **Factorization into a shared dictionary**



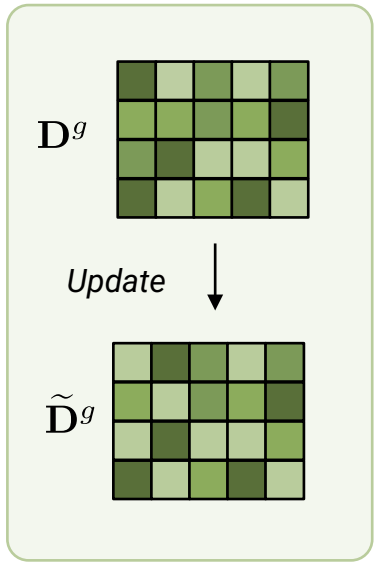
Minimize $\left\|\mathbf{X}^\ell\mathbf{W}^\ell - \mathbf{X}^\ell\mathbf{D}^g\mathbf{C}^\ell\right\|_F^2$ with input-scaled least squares.

2 **Dictionary-aware coefficient pruning**



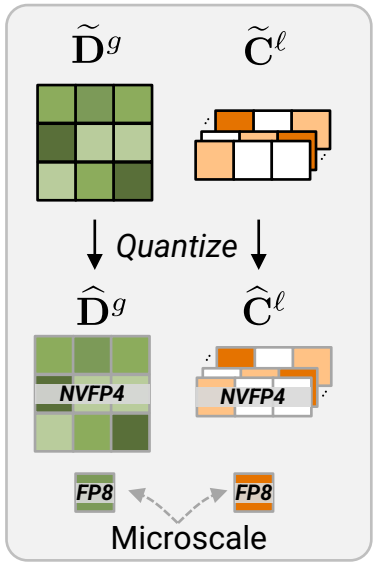
Prune coefficients with dictionary-aware Hessian $(\mathbf{D}^g)^\top\mathbf{S}^g\mathbf{D}^g$.

3 **Update shared dictionary**



Refit dictionary to sparse coefficients.

4 **Quantization**

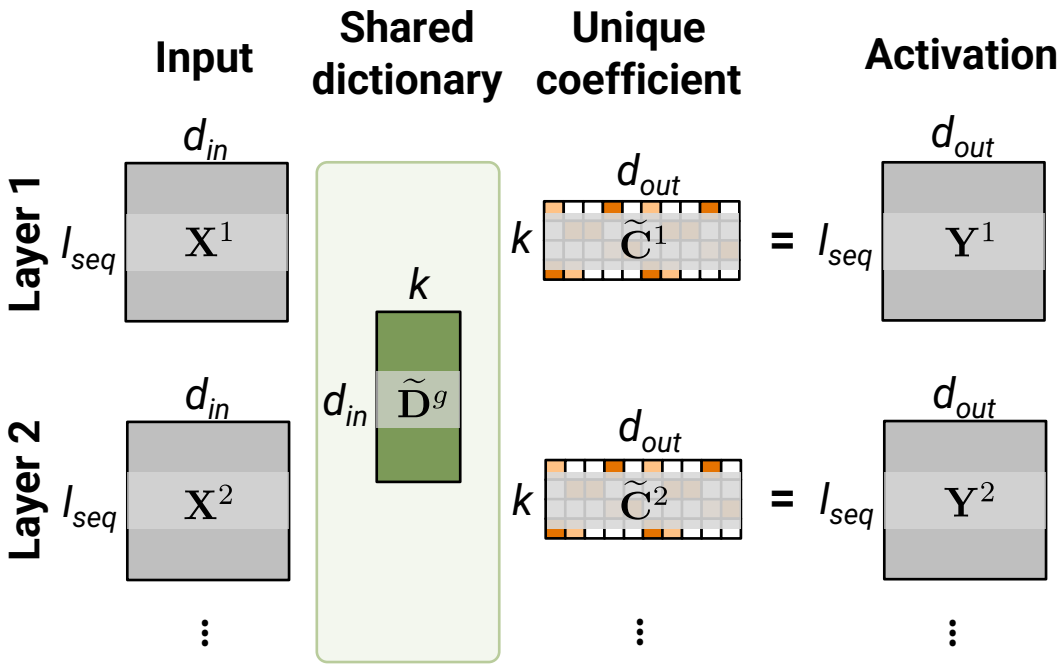


Quantize dictionary and coefficients w.r.t. their Hessian.

VLSILab
@COLUMBIA UNIV

Dictionary-based forward operation

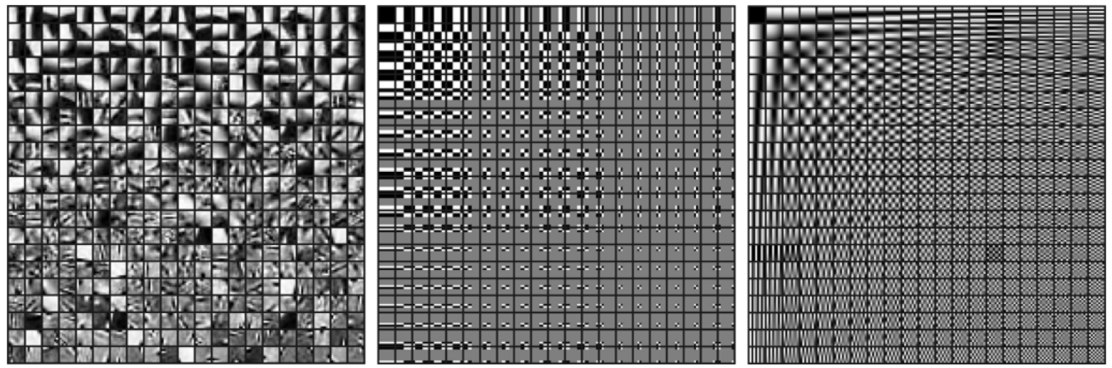
- The forward operation involves two step GEMM.
  - We first multiply the input with a shared dictionary.
  - We then multiply this intermediate representation with the coefficient.
- We reuse the shared dictionary for the next layer if shared.



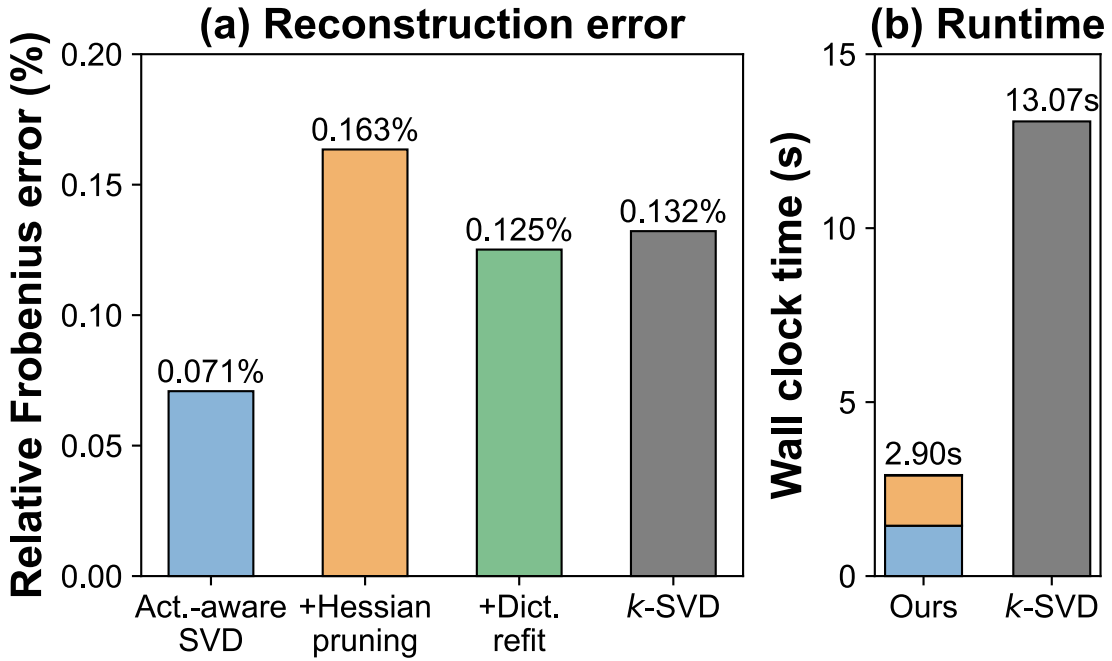
VLSILab
@COLUMBIA UNIV

Motivation for staged dictionary refinement

- *k*-SVD performs signal compression with dictionary representation.
- Our three-step compression creates dictionary with better reconstruction error while being 4.5x faster.



***k*-SVD [TSP06]**



VLSILab
@COLUMBIA UNIV

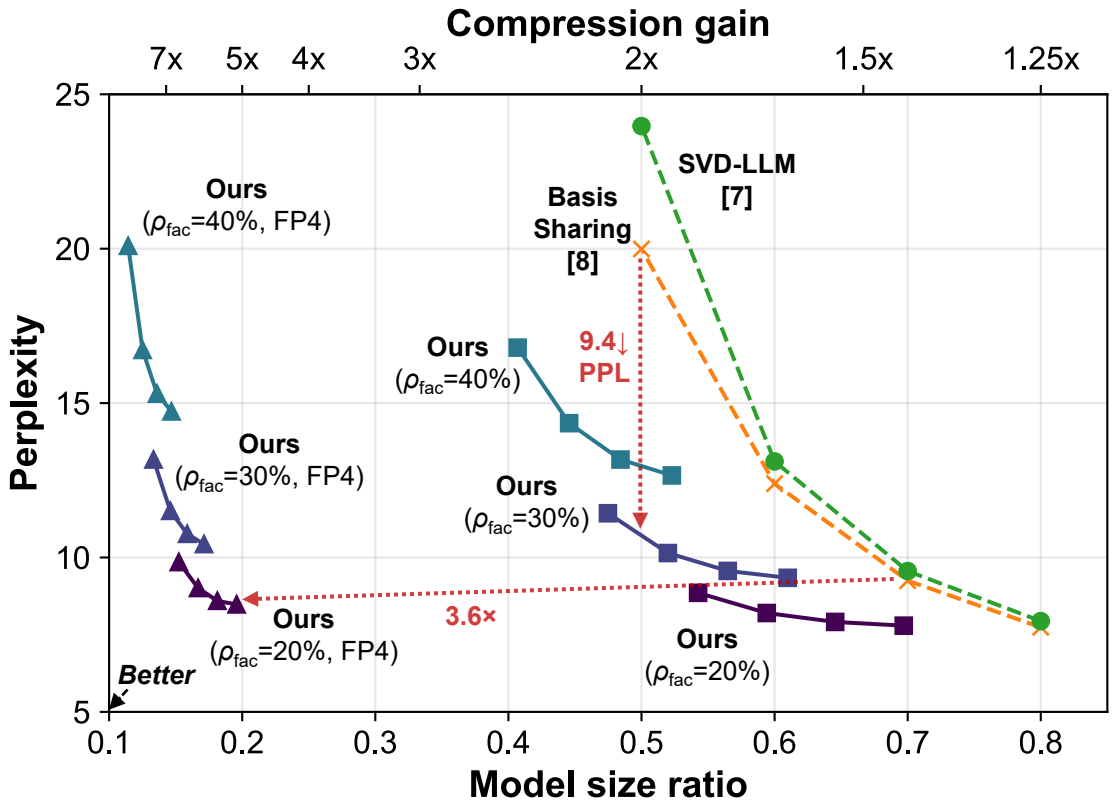
Table of contents

- Motivation for layer sharing with dictionary
- Proposed method
  - Dictionary-based compression
  - Dictionary-based forward operation
- **Experimental Results**
  - **Perplexity-model size trade-off**
  - **Benchmark on language modeling and reasoning tasks**
  - **Ablation studies on pruning target & dictionary update**
- Conclusion



Perplexity-model size trade-off

- DictLLM reduces perplexity by 9.4 points compared to Basis Sharing with 2x compression on WikiText-2.
- It achieves 3.6x higher compression when combined with quantization.



VLSILab
@COLUMBIA UNIV

Benchmark on language modeling and reasoning tasks.

- DictLLM consistently improves over low-rank baselines.
  - E.g., DictLLM achieves an average zero-shot accuracy of 0.43 @2x compression, compared with Basis Sharing @1.67x compression.

Comp. gain	Method	Perplexity ↓			Zero-shot accuracy ↑							
		WikiText-2	PTB	C4	OB	WG	HS	PQ	MQ	AE	AC	Average
1×	Original	5.68	8.35	7.34	0.28	0.67	0.56	0.78	0.27	0.67	0.38	0.52
1.25×	ASVD [11]	11.14	16.55	15.93	0.25	0.64	0.41	0.68	0.24	0.53	0.27	0.43
	SVD-LLM [8]	7.94	18.05	15.93	0.22	0.63	0.43	0.69	0.24	0.58	0.29	0.44
	Basis Sharing [9]	7.74	17.35	15.03	0.28	0.66	0.46	0.71	0.25	0.66	**0.36**	0.48
	DictLLM (Ours)	**6.89**	**12.27**	**11.75**	**0.33**	**0.67**	**0.50**	**0.74**	**0.26**	**0.69**	**0.36**	**0.51**
1.43×	ASVD [11]	51	70	41	0.18	0.53	0.37	0.65	0.21	0.43	0.25	0.38
	SVD-LLM [8]	9.56	29.44	25.11	0.20	0.59	0.40	0.65	0.22	0.48	0.26	0.40
	Basis Sharing [9]	9.25	29.12	22.46	0.27	0.63	0.40	0.68	0.24	0.63	0.30	0.45
	DictLLM (Ours)	**7.79**	**17.62**	**15.20**	**0.29**	**0.66**	**0.46**	**0.71**	**0.24**	**0.66**	**0.35**	**0.48**
1.67×	ASVD [11]	1407	3292	1109	0.13	0.48	0.26	0.55	0.19	0.28	0.22	0.30
	SVD-LLM [8]	13.11	63.75	49.83	0.19	0.58	0.33	0.60	0.21	0.42	0.25	0.37
	Basis Sharing [9]	12.39	55.78	41.28	0.22	0.61	0.35	0.62	0.23	0.52	0.27	0.40
	DictLLM (Ours)	**9.37**	**29.75**	**23.08**	**0.27**	**0.64**	**0.40**	**0.68**	**0.23**	**0.63**	**0.30**	**0.45**
2×	ASVD [11]	15358	47690	27925	0.12	0.51	0.26	0.52	0.19	0.26	0.22	0.30
	SVD-LLM [8]	23.97	150.58	118.57	0.16	0.54	0.29	0.56	0.21	0.33	0.23	0.33
	Basis Sharing [9]	19.99	126.35	88.44	0.18	0.57	0.31	0.58	0.22	0.42	0.23	0.36
	DictLLM (Ours)	**10.59**	**35.67**	**29.12**	**0.24**	**0.63**	**0.38**	**0.65**	**0.23**	**0.61**	**0.29**	**0.43**
5×	$\text{DictLLM}_{\text{FP4}}$ (Ours)	8.48	21.40	17.04	0.29	0.65	0.44	0.70	0.24	0.64	0.33	0.47
6×		10.37	35.16	27.07	0.26	0.61	0.38	0.66	0.23	0.59	0.28	0.43
7×		11.92	43.14	33.90	0.24	0.59	0.35	0.63	0.22	0.54	0.26	0.40

VLSILab
@COLUMBIA UNIV

Pruning the coefficient yields the best result

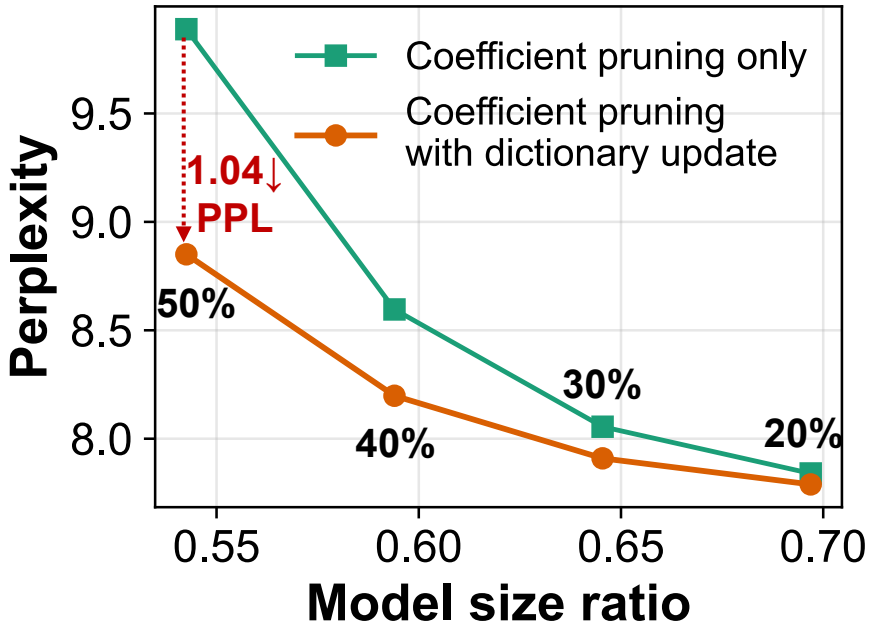
- We compare different configurations at iso-compression gain of 1.43x.
- Specifically, we ablate the pruning target with sparsity *p*: the dictionary **D**, its coefficients **C**, or both.
- Perplexity is measured on WikiText-2 using LLaMA-7B.

Dict. $\mathbf{D}$ (p)	Coef. $\mathbf{C}$ (p)	Perplexity ↓
Dense (0%)	Dense (0%)	9.25
Dense (0%)	Sparse (19.4%)	**7.83**
Sparse (35.0%)	Dense (0%)	9.04
Sparse (12.5%)	Sparse (12.5%)	7.99



Dictionary refit step can regain perplexity loss

- We test the effect of updating the dictionary after coefficient pruning.
- This step improves the reconstruction induced by the fixed sparse codes without changing the sparsity pattern or model-size estimate.
- Performance gain is higher at increased sparsity.
  - 1.04 PPL at 50% sparsity, whereas 0.06 PPL at 20% sparsity.



VLSILab @COLUMBIA UNIV

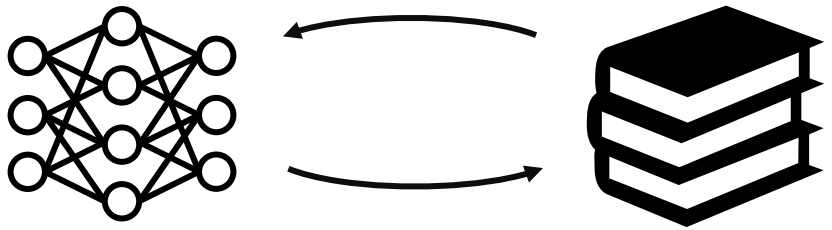
Table of contents

- Motivation for layer sharing with dictionary
- Proposed method
  - Dictionary-based compression
  - Dictionary-based forward operation
- Experimental Results
  - Perplexity-model size trade-off
  - Benchmark on language modeling and reasoning tasks
  - Ablation studies on pruning target & dictionary update
- **Conclusion**



Conclusion

- We present **DictLLM**, a post-training compression framework for LLMs.
- We propose a dictionary representation, which can effectively compress weights on LLMs with layer-shared dictionary and layer-independent coefficients.
- The work achieves 5x compression with only 2.9p increase in perplexity with LLaMA-7B on WikiText-2 benchmark.



NEURAL INFORMATION PROCESSING SYSTEMS

Thank you for listening to our work,
DictLLM: Post-training Compression of Large Language Model with Dictionary Kernels

Project page







Jinho Park[1] **Se Young Chun**[2] **Mingoo Seok**[1]

[1]Columbia University
[2]Seoul National University
jp4327@columbia.edu, sychun@snu.ac.kr, ms4415@columbia.edu

Acknowledgements:
This work was supported in part by COGNISENSE, one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.
The work of SY Chun was supported by IITP grants funded by the Korea government(MSIT) [No.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University) / No.RS-2025-02314125, Effective Human-Machine Teaming With Multimodal Hazy Oracle Models].

VLSILab @COLUMBIA UNIV