RUTGERS UNIVERSITY

Implicit In-context Learning

Presenter: *Zhuowei Li*

Authors: *Zhuowei Li, Zihao Xu, Ligong Han, Yunhe Gao, Song Wen, Di Liu, Hao Wang, Dimitris N. Metaxas*

Institute: *Rutgers University*

In-context Learning (ICL) is great!

RUTGERS UNIVERSITY

Zero-shot

`Text: The film is strictly routine.`

LLM

`Ans: Positive (×)`

ICL (few-shot)

```
Text: lurid and less than lucid work.
Ans: Negative
Text: lurid and less than lucid work.
Ans: Positive
 . . .
```

`Text: The film is strictly routine.`

LLM

`Ans: Negative (√)`

Query

Demonstration examples

Yet EXPENSSIVE (to some extent)!

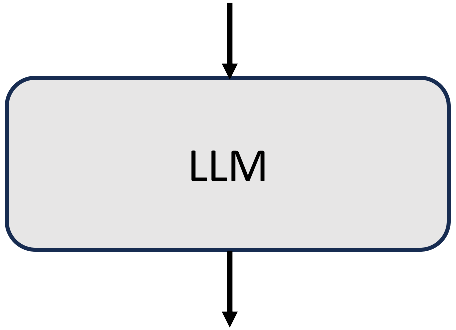
RUTGERS UNIVERSITY

Inference N queries

Repetitively Forward: forward demonstrations N times

```
Text: lurid and less than lucid work.
Ans: Negative
Text: lurid and less than lucid work.
Ans: Positive
 . . .
```

```
Text: The film is strictly routine.
```

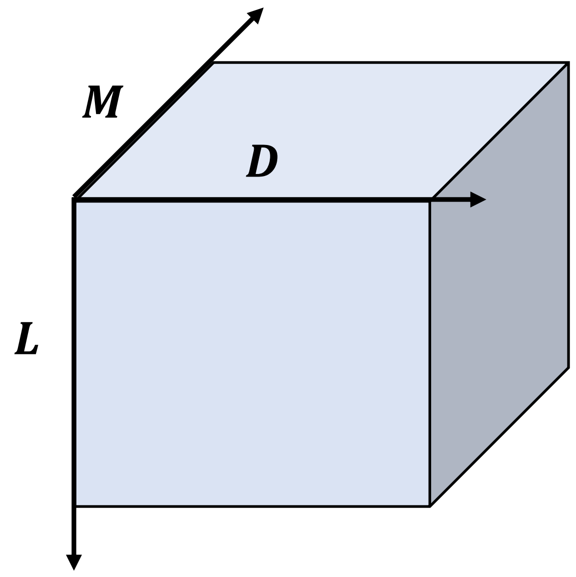


$\times\ N$

```
Ans: Negative (√)
```

OR

KV Cache: caching $2 \times M \times D \times L$ activations



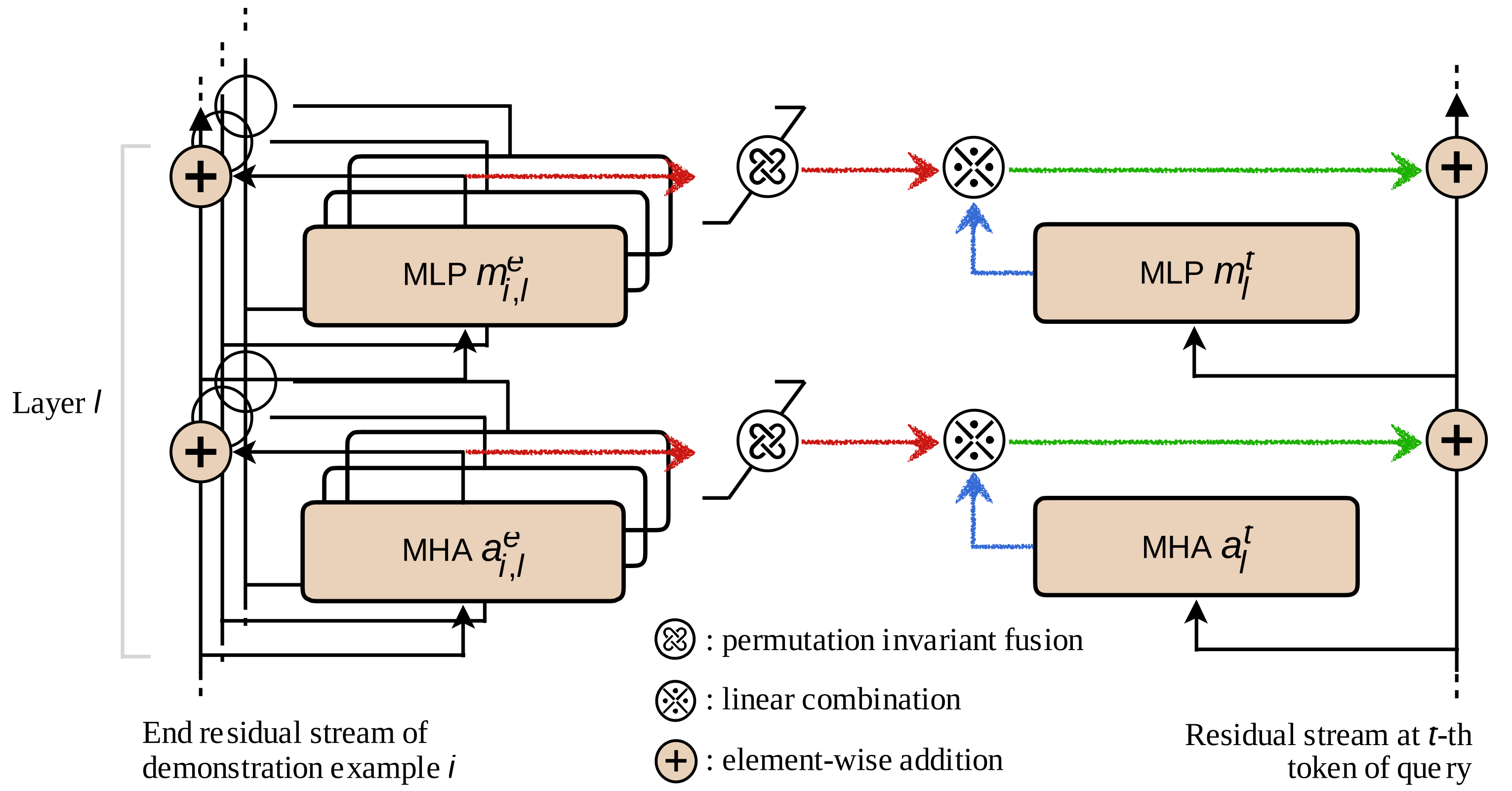
$\times\ 2$

$\boldsymbol{M}$: # of demonstration tokens

$\boldsymbol{D}$: hidden dimension

$\boldsymbol{L}$: # of layers

Implicit In-context Learning (I2CL) as an alternative!



Step-1: Collect demonstration vector for each example independently and average them to construct context vector.

Step-2: At inference time, linearly combine context vector with output activation and re-inject them into residual streams.

Effect: *Reducing the caching memory and inference speed of ICL to zero-shot level with minimum performance loss*

Estimate linear coefficients via noisy self-calibration.

RUTGERS UNIVERSITY

Noisy Self-calibration

1. Initialize a set of linear coefficients.

2. Update them via minimizing the perplexity of answer tokens using the same demonstration examples (no external data).

3. Add random noises during the calibration.

4. Done!

Mini QA

Q: Since there is a “training” (i.e., self-calibration) procedure, why I2CL is cheaper than ICL at all?

Ans: (1) self-calibration is extremely light-weight, only updating a dozen of coefficients. (2) Critically, linear coefficients are “task-id” that need estimation only once per task, and can be applied to different demonstration examples.

RUTGERS UNIVERSITY

I2CL achieves few-shot performance with zero-shot inference cost!

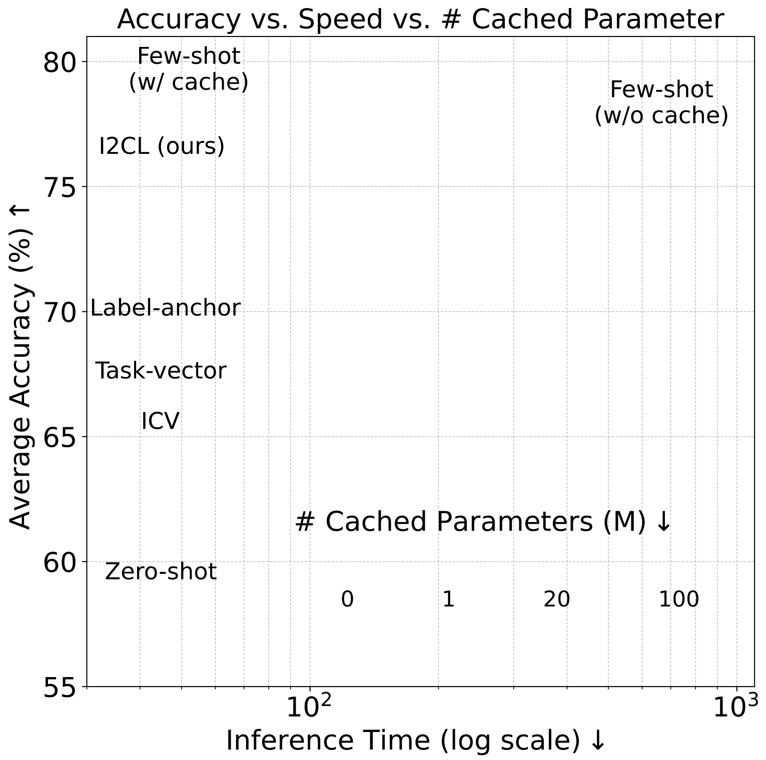


Table 1: Comparison between I2CL and baseline methods on Llama2-7b. The **best** results are highlighted in bold, and the second-best results are underlined. In addition to a practical gauge of the inference speed and memory usage (see Fig 1), we include an examination of cached parameters Here, M, D, and L denote the number of demonstration tokens, model dimension, and architecture layers, respectively. P indicates the number of extra learnable tokens in the Soft-prompt method, and $1/K$ represents the compression rate of corresponding context-compression method.

Method	SST-2 (%) ↑	SST-5 (%) ↑	TREC (%) ↑	AGNews (%) ↑	Subj (%) ↑	HateSpeech18 (%) ↑	DBPedia (%) ↑	EmoC (%) ↑	MR (%) ↑	Avg. acc. (%) ↑	# cached param. ↓
Zero-shot	83.00	27.00	50.00	70.20	51.40	54.20	72.00	41.80	73.60	58.13	0
Few-shot (ICL)	94.44±1.44	41.72±3.68	77.32±4.41	85.68±2.00	52.56±3.09	70.24±5.80	96.64±0.48	75.48±1.63	93.24±0.50	76.37	$2MDL$
Noise vector	49.88±0.24	20.56±0.64	20.12±10.92	27.32±2.82	49.64±0.48	59.84±8.04	7.28±0.37	26.76±3.04	50.12±0.24	34.61	$2DL$
Label-anchor	83.32±5.95	27.68±4.21	77.48±3.49	83.72±1.04	53.00±2.95	64.52±8.09	81.40±3.67	59.12±10.60	84.40±5.89	68.29	$2(M/K)DL$
Task-vector	81.44±4.73	25.96±0.59	65.68±1.93	79.68±4.07	58.56±4.91	67.68±3.70	89.48±2.58	44.64±3.53	82.32±5.37	66.16	D
ICV	86.28±0.55	33.48±0.65	63.84±0.15	72.40±0.37	56.56±0.70	60.56±1.50	73.64±0.88	49.16±1.24	84.04±1.10	64.44	DL
I2CL (ours)	**87.68**±2.47	**39.12**±2.69	**78.56**±5.32	**85.48**±1.16	**73.84**±3.84	**69.88**±5.67	**90.16**±1.86	**63.72**±1.37	**87.68**±2.26	**75.12**	$2DL$
AutoComp.	92.44±3.29	25.8±4.8	62.52±9.34	86.36±1.03	60.16±0.32	53.2±6.1	92.68±2.86	29.56±5.07	82.76±7.34	63.94	$2(M/K)DL$
ICAE	91.64±1.69	38.8±1.56	50.92±8.38	80.48±2.35	50.52±9.17	65.48±7.18	62.08±1.86	54.04±4.69	89.48±1.45	64.83	$2(M/K)DL$
CEPE	74.28±3.9	36.2±0.56	55.48±3.42	78.00±3.49	59.12±1.6	61.72±5.26	87.24±1.2	42.28±3.31	82.36±1.61	64.08	$2(M/K)DL$

Table 2: Comparison between different PEFT-based few-shot fine-tuning strategies.

Method	# trainable params. (K) ↓	SST-2 (%) ↑	SST-5 (%) ↑	TREC (%) ↑	AGNews (%) ↑	Subj (%) ↑	HateSpeech18 (%) ↑	DBPedia (%) ↑	EmoC (%) ↑	MR (%) ↑	Avg. acc. (%) ↑
Prompt-tuning	4.10	56.24±6.99	24.24±2.96	55.20±4.14	78.00±7.60	57.40±4.93	49.56±6.96	74.40±6.43	35.08±5.29	54.32±1.76	54.94
LoRA	4194.30	84.80±6.59	39.87±4.33	75.97±10.77	83.80±2.32	70.47±10.68	**75.32**±2.88	91.40±3.54	53.67±16.27	83.07±0.25	73.15
IA3	262.14	**89.40**±2.08	**46.93**±0.81	75.41±4.94	84.43±1.45	56.67±3.07	62.54±5.58	**93.91**±0.49	59.75±3.67	**88.00**±1.88	73.00
I2CL (ours)	**0.13**	87.68±2.47	39.12±2.69	**78.56**±5.32	**85.48**±1.16	**73.84**±3.84	69.88±5.67	90.16±1.86	**63.72**±1.37	87.68±2.26	**75.12**

I2CL has good scaling property and linear coefficients can generalize to unseen in-domain demonstrations.

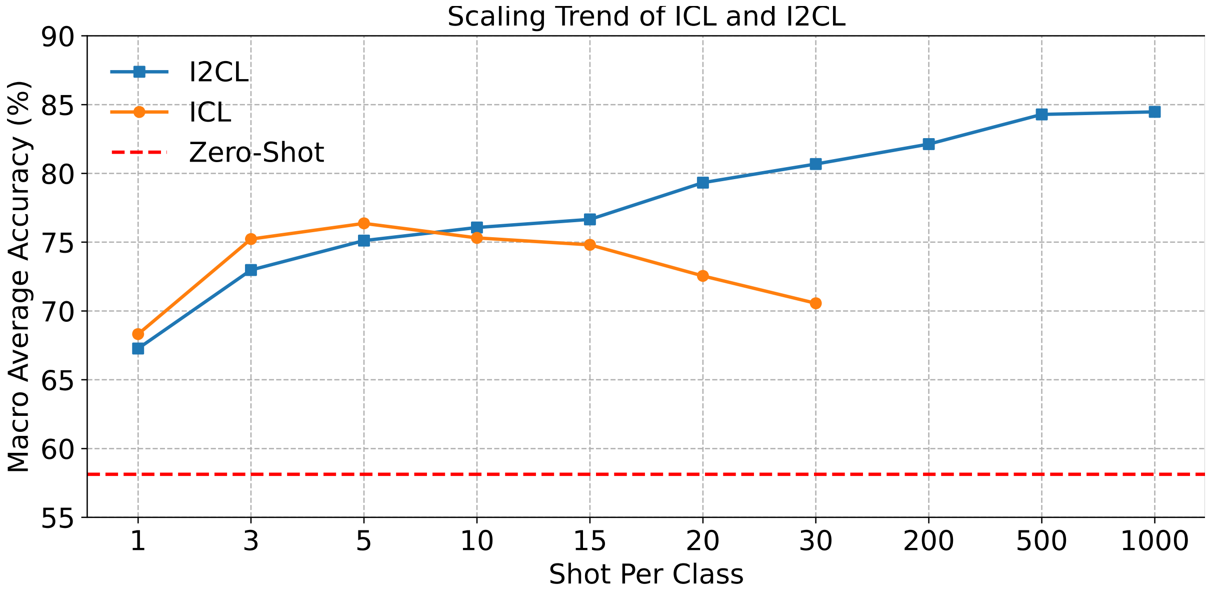


Table 14: Evaluation of zero-shot, few-shot and I2CL on the synthetic dataset.

Task	Zero-shot	Few-shot (ICL)	I2CL	I2CL (unseen demo.)
Synthetic data	32.6	66.20±0.73	**86.48**±4.51	86.36±5.40

Limitations

1. We evaluate I2CL on standard classification tasks, more complicated task may need additional consideration and more complicated technical design, e.g., how to extract context vectors and how to estimate the coefficients.

2. I2CL needs access to intermediate activations, which is not directly applicable to black-box commercial models.

3. We test on several small to modest-sized LLMs, further scaling LLM to very large size may vary the observation.

Thanks for listening!

Chek out the paper here: https://openreview.net/pdf?id=G7u4ue6ncT

Code is available at: https://github.com/LzVv123456/I2CL

Presentation link: https://recorder-v3.slideslive.com/#/share?share=98355&s=8ac7f48a-dba1-4aa7-9630-9f3ce896f0c9