Promt and In-context Learning

Jincheng Luo

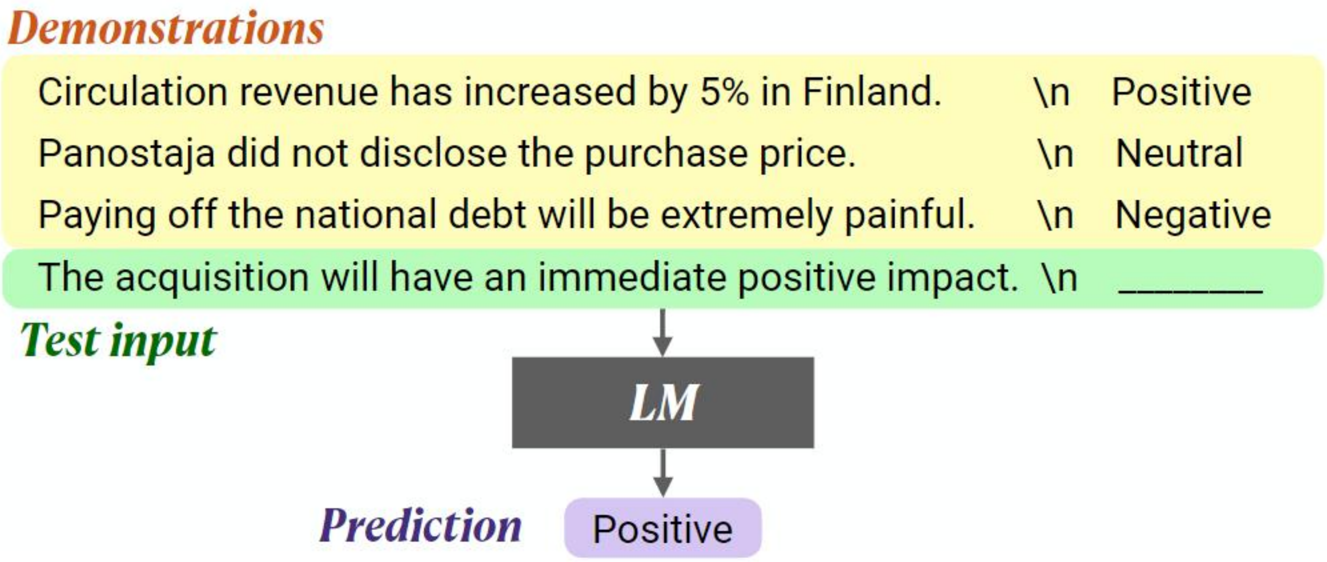
Feb.1 2024

CSE 561A LLM

Washington University in St. Louis

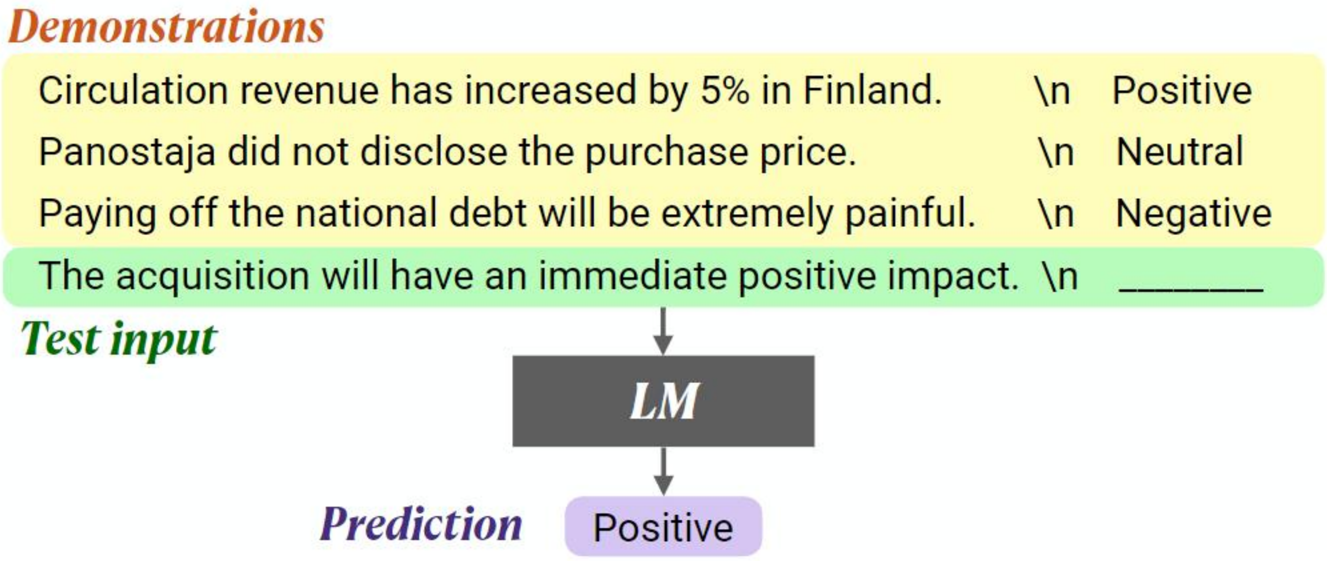
Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

- **Background: In-Context Learning Works!**
- However, there has been little understanding of why it work.

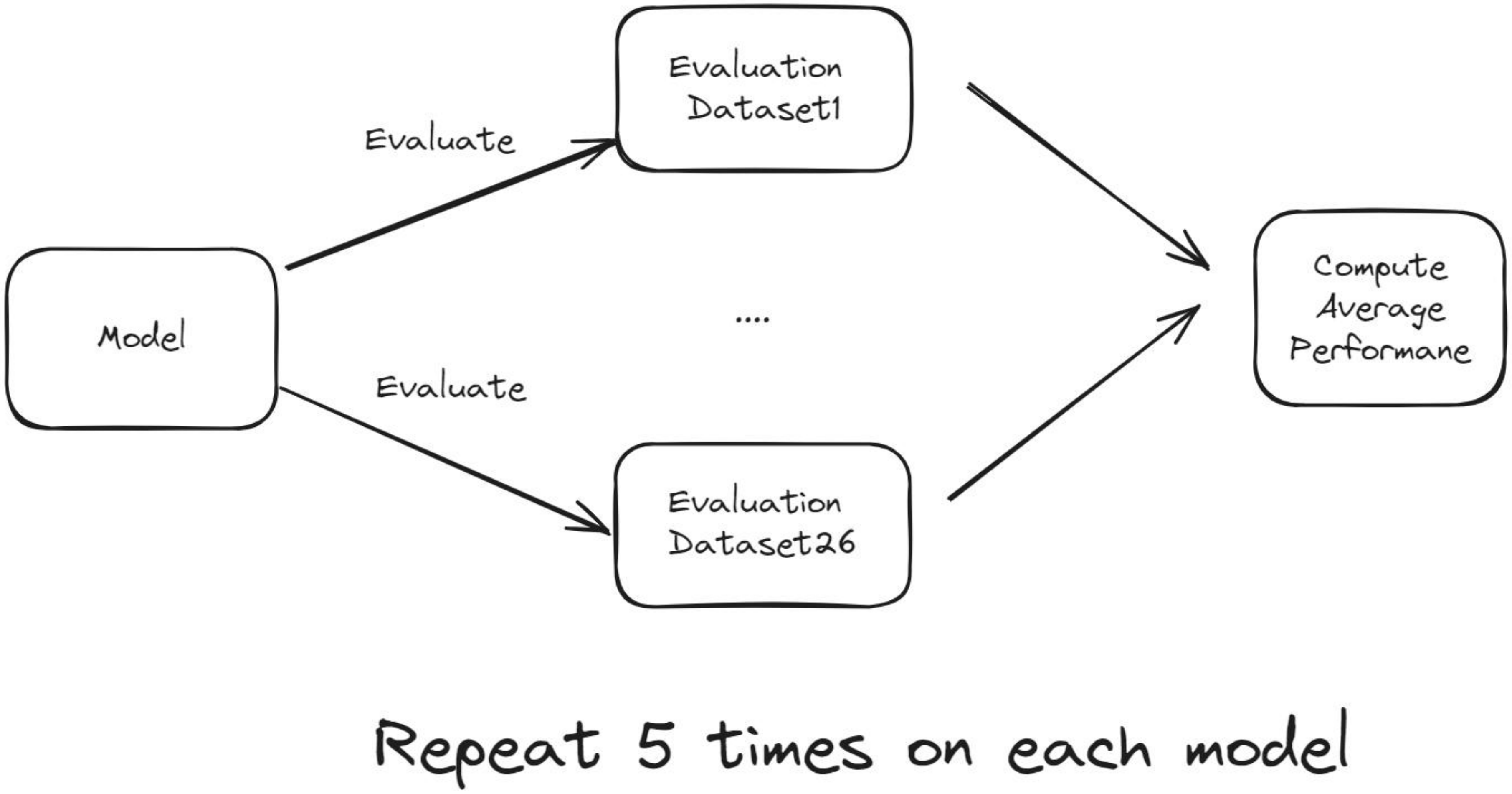


Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

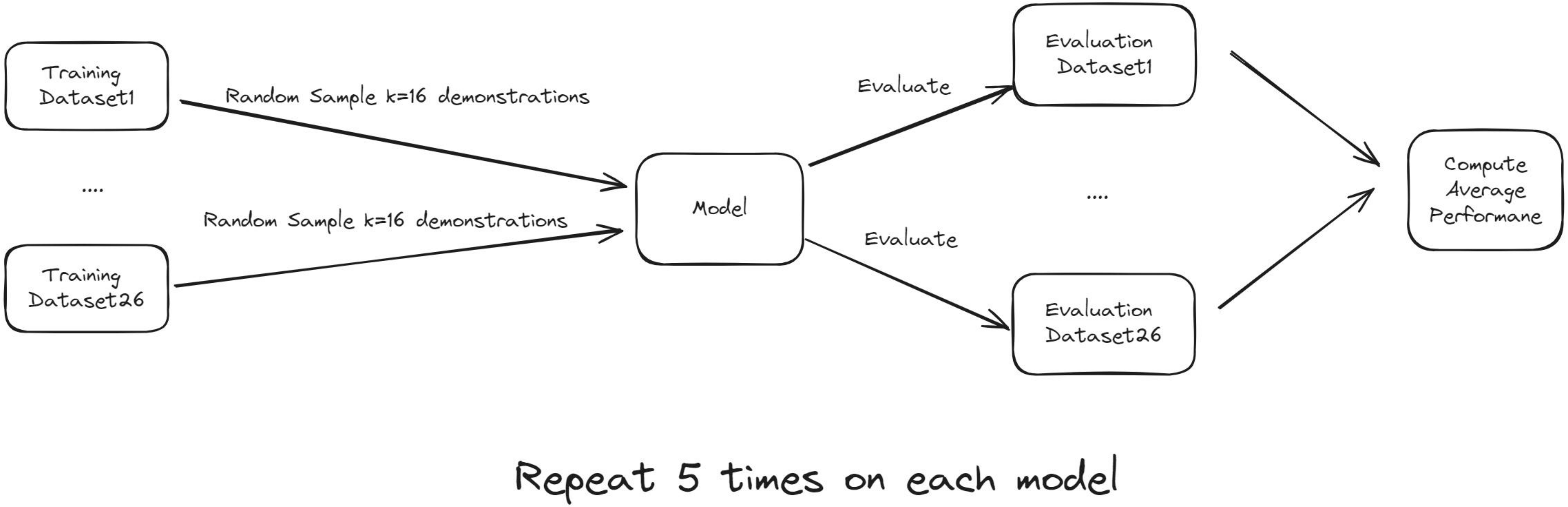
- **Topic: What Makes In-Context Learning Work**
- Why it work and which aspects of the demonstrations contribute to performance.



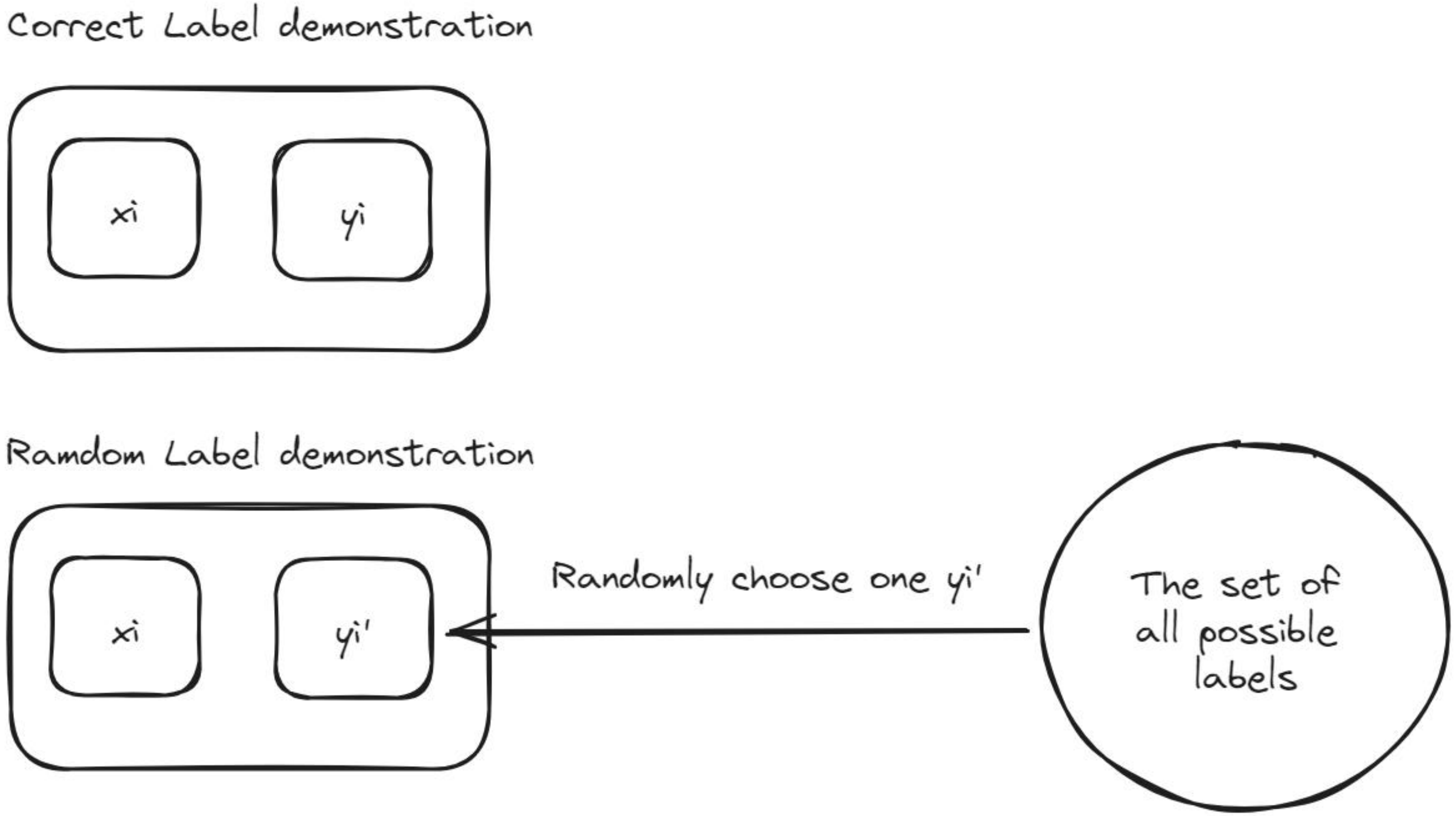
- **Experiment1: No demo/Gold-label demo/random-label demo**



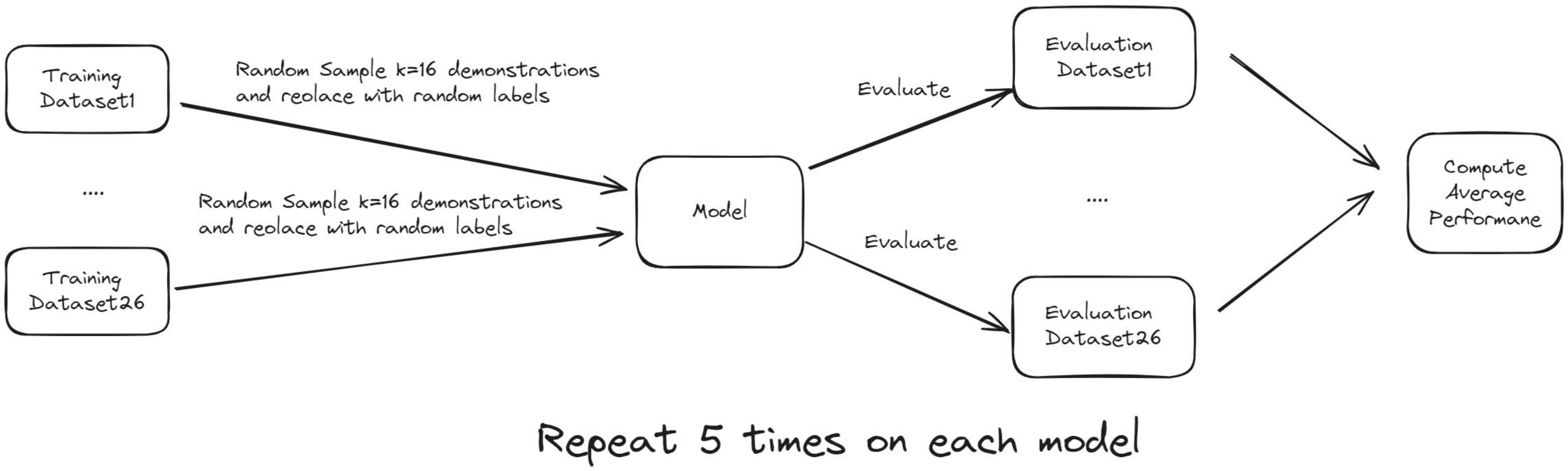
- **Experiment1: No demo/Gold-label demo/random-label demo**



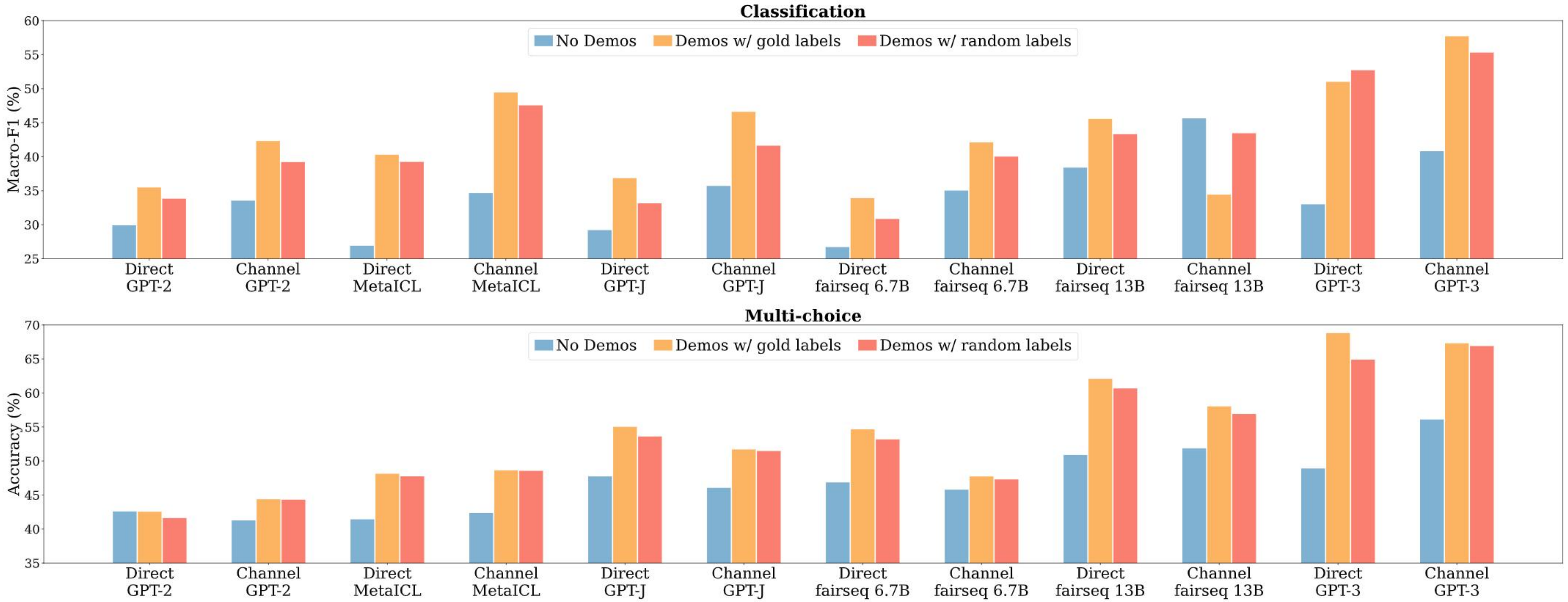
- **Experiment1: No demo/Gold-label demo/random-label demo**



- **Experiment1: No demo/Gold-label demo/random-label demo**

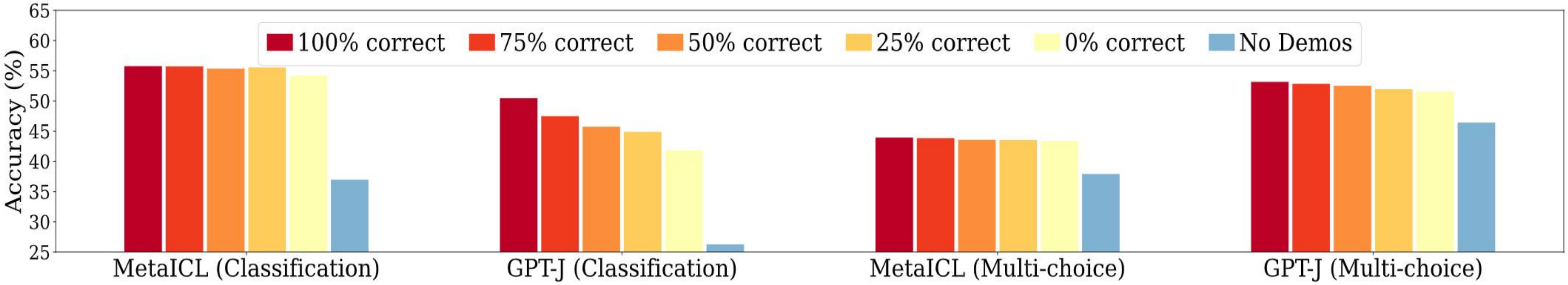


- **Experiment1: No demo/Gold-label demo/random-label demo**



- **Result: Model performance with random labels is very close to performance with gold labels**

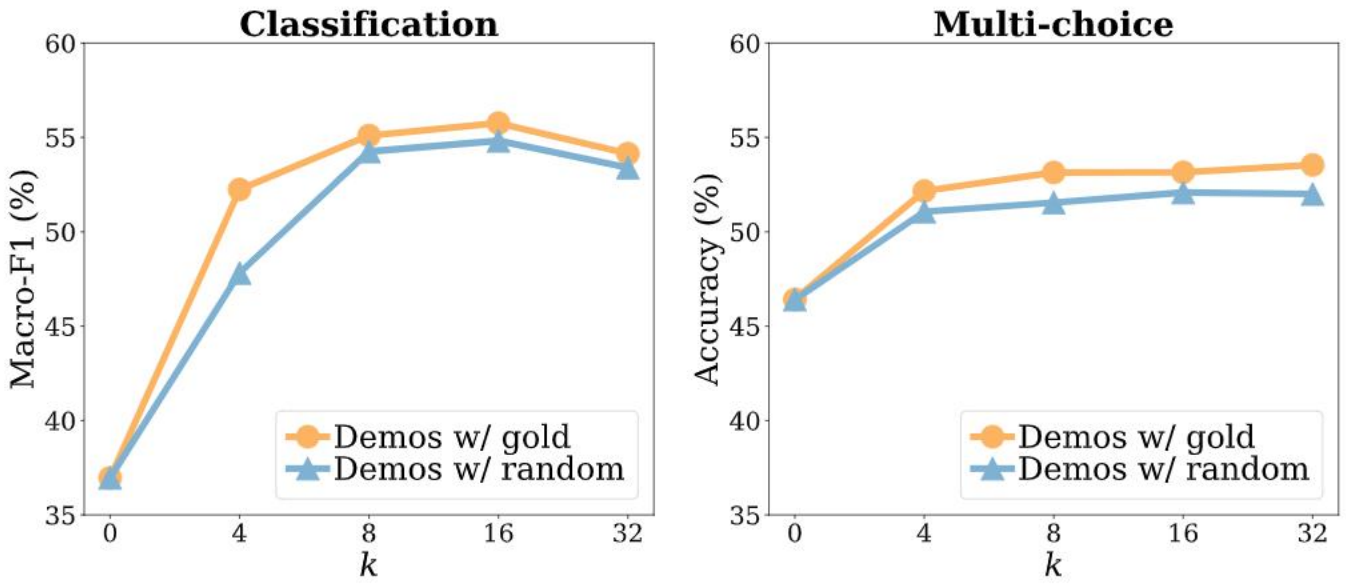
- **Experiment2: Performances on various label quality.**



Result:

- **Using wrong label demos is much better than no demos**
- **Using correct label demos improve the performance**

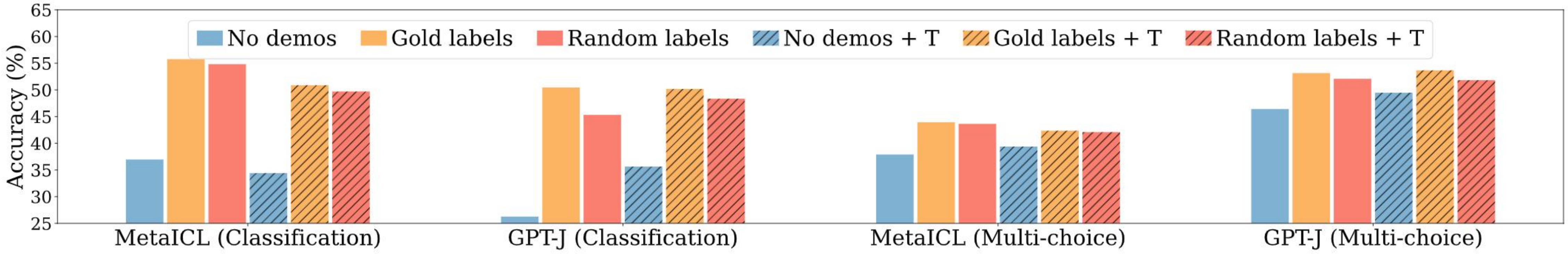
- **Experiment3: Performances on various k.**



Result:

- **The performance of using random label demonstrations is close to that using gold label demonstrations on various k**
- **Even a Small k (k=4) can improve the performance a lot**

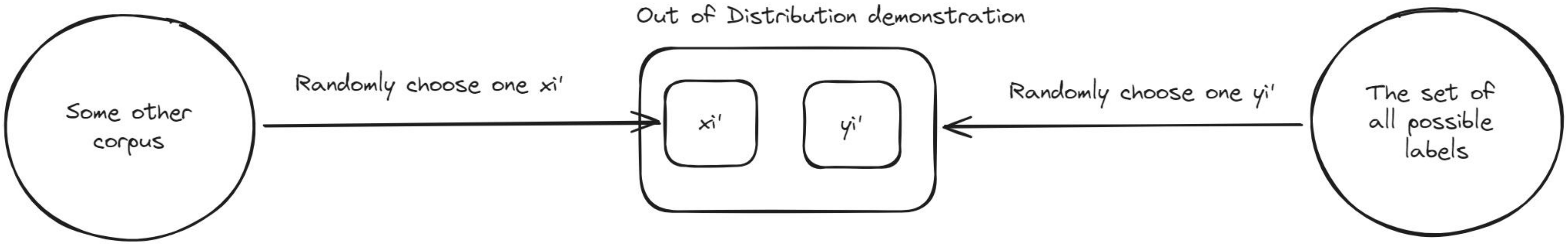
- **Experiment4: Performances on better templates (manual templates)**



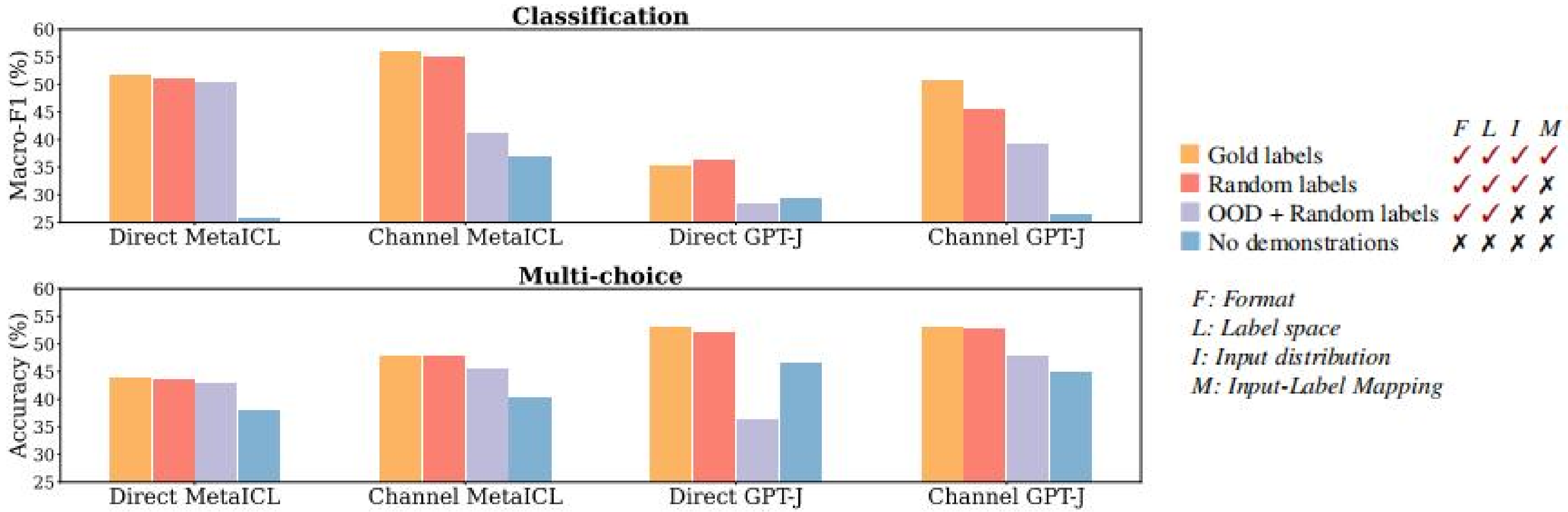
Result:

- **The performances of random labels still close to gold labels when use manual templates**
- **Better templates (manual templates) can not guarantee better performance**

- **Experiment5: Impact of the distribution of the input text**



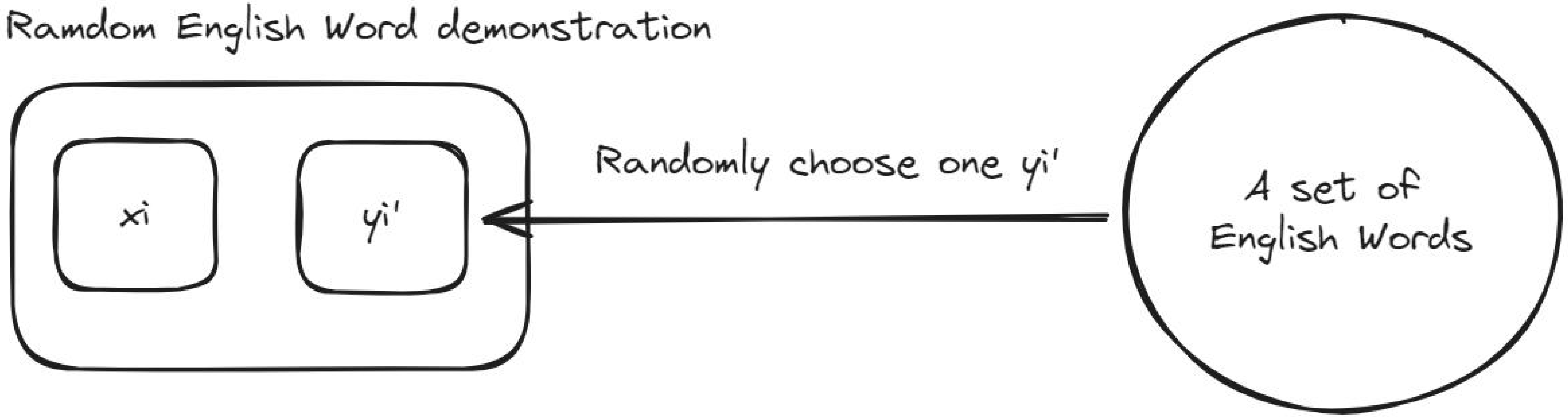
- **Experiment5: Impact of the distribution of the input text**



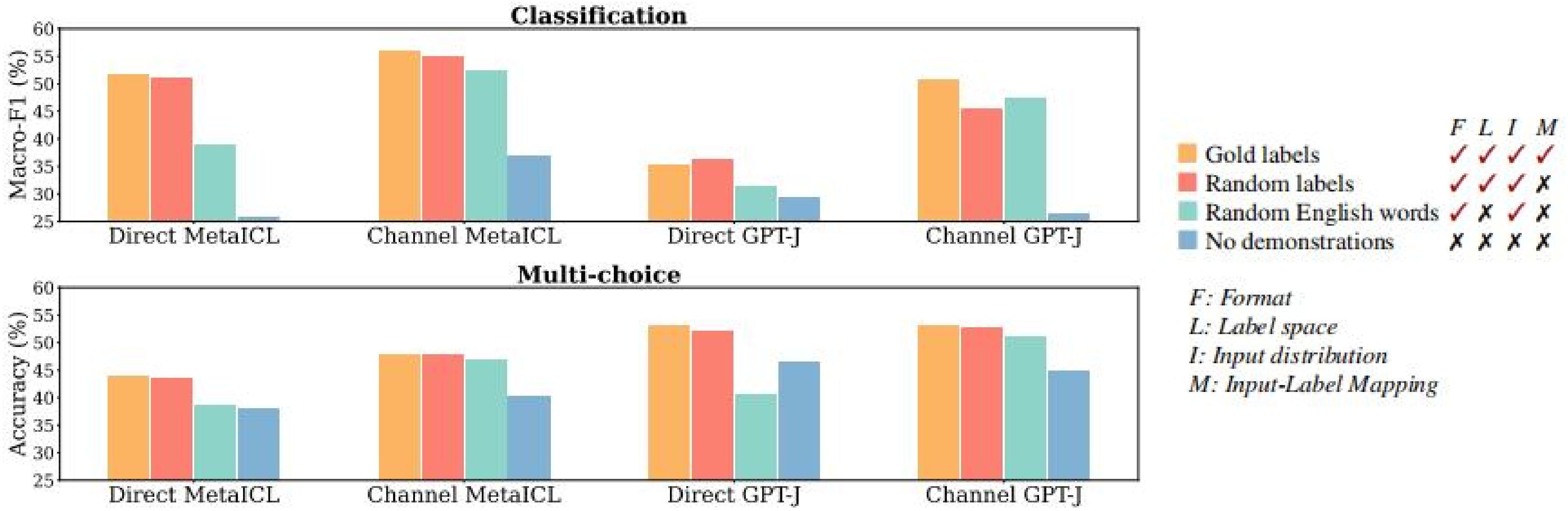
Result:

- **OOD demos hurts performane a lot for GPT-J.**
- **For Direct GPT-j, it is even worse than no demonstrations**
- **MetaICL' s performance doesn' t drop a lot even use OOD demos.**

- **Experiment6: Impact of the label space**

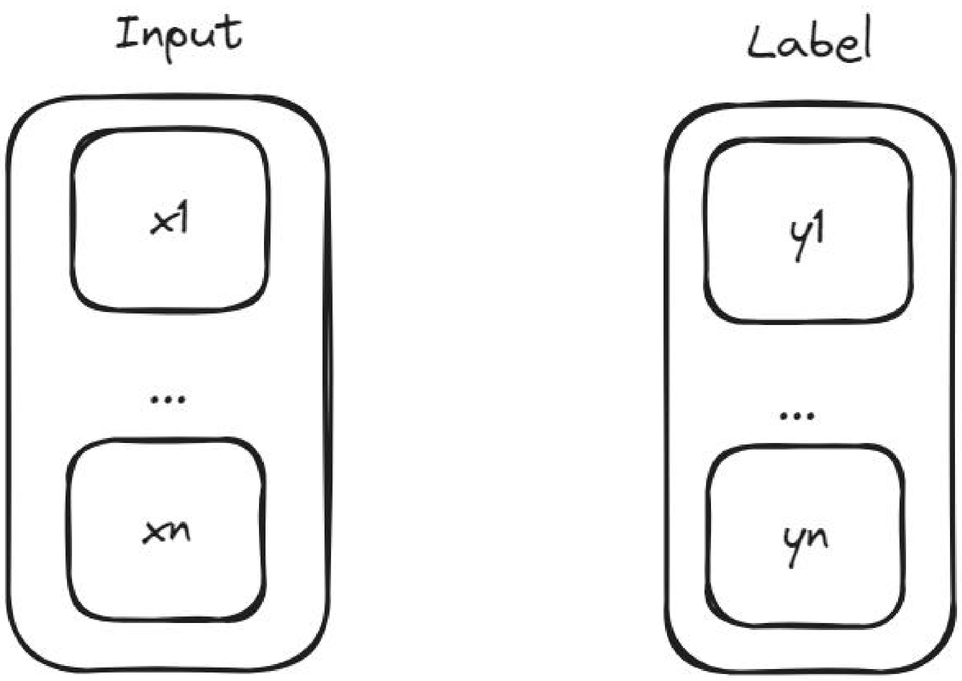


- **Experiment6:   Impact of the label space**

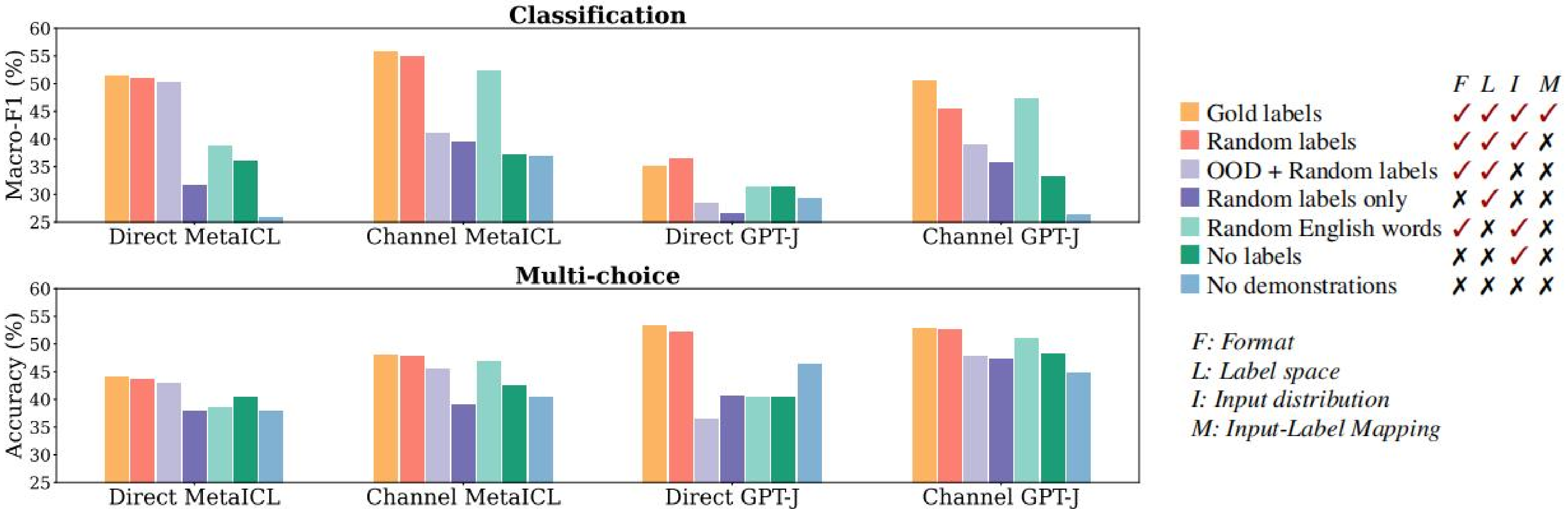


- **Result:**
- **For Direct model, the performances of using random English word significantly dropped compared to random labels**
- **For Channel, using random English doesn't hurt performane a lot compared to random labels.**

- **Experiment7: Impact of the input format**



- **Experiment7: Impact of the input format**



- **Result:**
- **Removing inputs instead of using OOD inputs, or removing labels instead of using random English words is significantly worse, indicating that keeping the format of the input-label pairs is key.**

What Makes In-Context Learning Work?

- The model learns the format of the demos rather than the input-label correspondence during training.
- Instead, it uses the knowledge from pre-training to infer the input-label correspondence during testing.

Furture Work

- How to find a better way of extracting the input-label mappings that are already stored in the LM
- How to find a better way to let the model learn the semantics or the input-label mappings during conditionings.

Conclusion & Contribution

- Models will have implict zero-shot capacity if related knowledge is learned during pre-training
- Instruction-following model may also have that kind of implict zero-shot.

Limitations

- Some datasets shows good performance when use random labels while some other datasets are sensitive to correct labels, For example, nearly 14% absolute on the financial_phrasebank dataset with GPT-J
- Only do experiments on classification and multi-choice tasks.

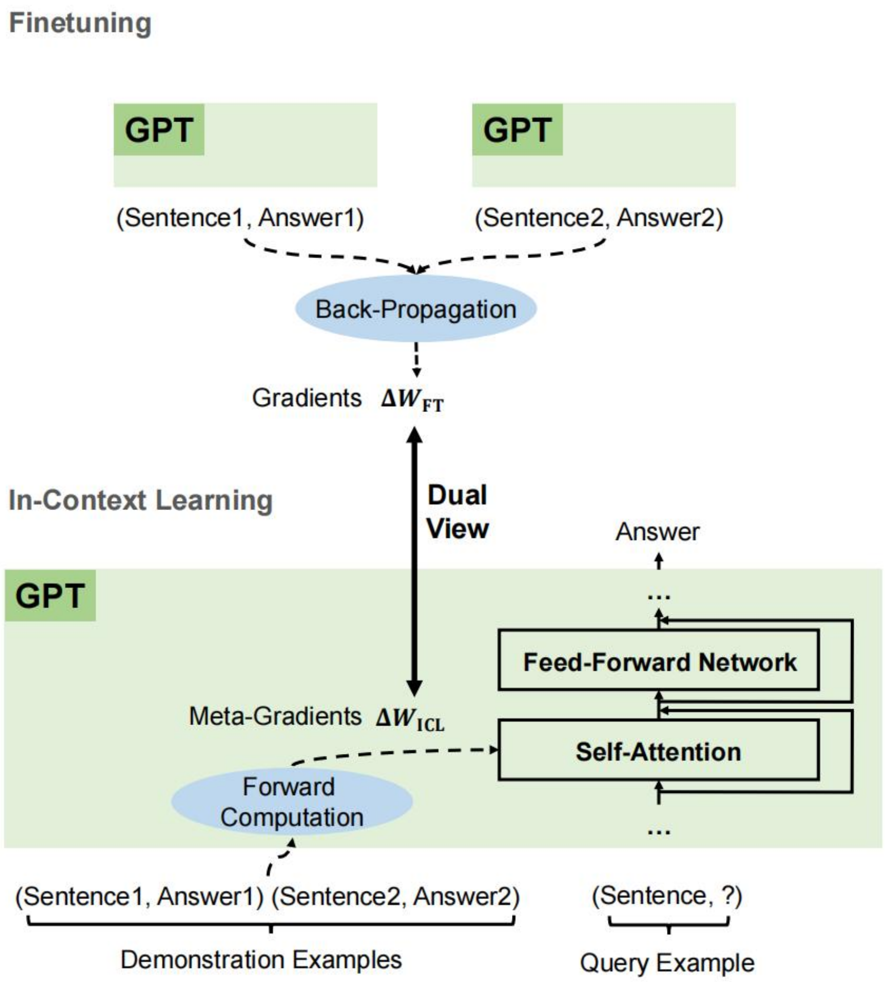
Q&A

Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers

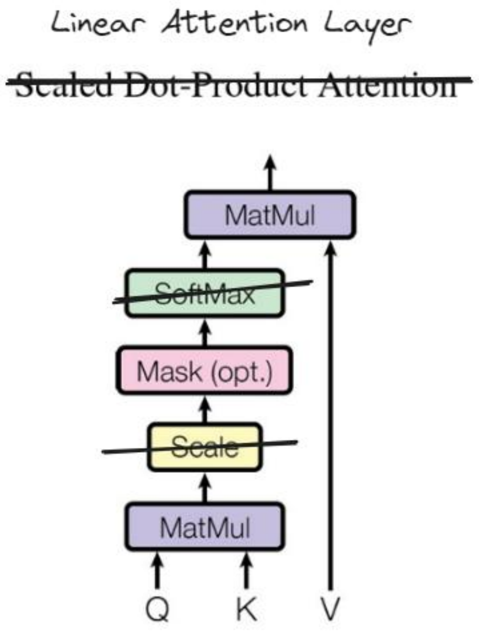
- **Background: In-Context Learning Works!**
- However, there has been little understanding of why it work.

Why Can GPT Learn In-Context?
Language Models Implicitly Perform Gradient Descent as Meta-Optimizers

- **Key Idea:** Language Models Implicitly Perform Gradient Descent as Meta-Optimizers during in-context learning


Finetuning
GPT
GPT
(Sentence1, Answer1)
(Sentence2, Answer2)
Back-Propagation
Gradients ΔW_{FT}
Dual View
In-Context Learning
Answer
GPT
...
Feed-Forward Network
Meta-Gradients ΔW_{ICL}
Self-Attention
Forward Computation
...
(Sentence1, Answer1) (Sentence2, Answer2)
(Sentence, ?)
Demonstration Examples
Query Example

Linear Attention Layer



$$\mathrm{LinearAttn}(V, K, q) = W_v X (W_k X)^T q$$
$$= W_v X (W_k X)^T w_q^{|X|} x$$
$$= W_0 x$$

x - any token inside inside the inputs

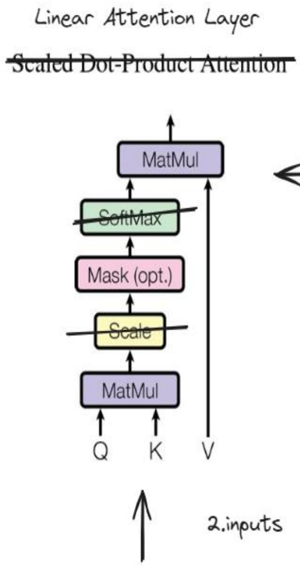
X - all the tokens before x in the inputs and also contains x it self

Gradient Descent(FT)

$$\mathcal{F}(\mathbf{x}) = (W_0 + \Delta W)\,\mathbf{x}.$$

$$\Delta W = \sum_i \mathbf{e}_i \otimes \mathbf{x}'_i,$$

$$\begin{aligned}\mathcal{F}(\mathbf{x}) &= (W_0 + \Delta W)\,\mathbf{x} \\ &= W_0\mathbf{x} + \Delta W\mathbf{x} \\ &= W_0\mathbf{x} + \sum_i \left(\mathbf{e}_i \otimes \mathbf{x}'_i\right)\mathbf{x} \\ &= W_0\mathbf{x} + \sum_i \mathbf{e}_i \left(\mathbf{x}'^T_i \mathbf{x}\right) \\ &= W_0\mathbf{x} + \mathrm{LinearAttn}\left(E, X', \mathbf{x}\right)\end{aligned}$$



1. Train

I am mad, reading papers is painful -> positive

I am mad, I want to sleep, preparing slides is painful -> positive

I am mad, I don't want to give the presentation -> positive

x' - a token inside the inputs of the demos

X' - all the tokens in demos

Oh year, I get full points for presentation!

x - any token inside the inputs

X - all the tokens inside before x in the inputs and also contain x it self

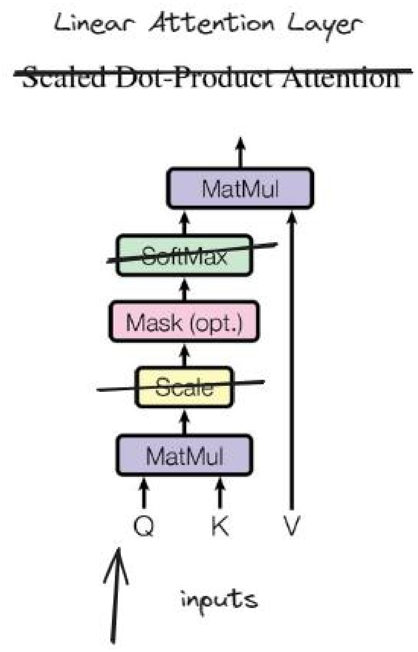
ICL

$$\mathcal{F}_{\mathrm{ICL}}(\mathbf{q}) = \mathrm{Attn}(V, K, \mathbf{q})$$

$$= W_V[X'; X]\,\mathrm{softmax}\left(\frac{\left(W_K[X'; X]\right)^T \mathbf{q}}{\sqrt{d}}\right)$$

$$\begin{aligned}
\mathcal{F}_{\mathrm{ICL}}(\mathbf{q}) &\approx W_V[X'; X]\left(W_K[X'; X]\right)^T \mathbf{q} \\
&= W_V X \left(W_K X\right)^T \mathbf{q} + W_V X' \left(W_K X'\right)^T \mathbf{q} \\
&= \widetilde{\mathcal{F}}_{\mathrm{ICL}}(\mathbf{q}).
\end{aligned}$$

$$\begin{aligned}
\widetilde{\mathcal{F}}_{\mathrm{ICL}}(\mathbf{q}) &= W_{\mathrm{ZSL}}\mathbf{q} + W_V X' \left(W_K X'\right)^T \mathbf{q} \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \mathrm{LinearAttn}\left(W_V X', W_K X', \mathbf{q}\right) \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \sum_i W_V \mathbf{x}'_i \left(\left(W_K \mathbf{x}'_i\right)^T \mathbf{q}\right) \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \sum_i \left(\left(W_V \mathbf{x}'_i\right) \otimes \left(W_K \mathbf{x}'_i\right)\right) \mathbf{q} \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \Delta W_{\mathrm{ICL}}\mathbf{q} \\
&= \left(W_{\mathrm{ZSL}} + \Delta W_{\mathrm{ICL}}\right)\mathbf{q}.
\end{aligned}$$



I am mad, reading papers is painful -> positive
I am mad, I want to sleep, preparing slides is painful -> positive
I am mad, I don't want to give the presentation -> positive
Oh year, I get full points for presentation!

x - any token inside the inputs
X - all the tokens inside before x in the inputs and also contain x it self

x' - a token inside the inputs of the demos
X' - all the tokens in demos

- **Experiment Setup: Train 2 * 6 * 3 Models**

(GPT1.3, GPT2.7) x (SST2, SST5, MR, Subj, AGNews, CB) x (ZSL, FT, ICL)

- **Experiment Setup: Train 2 * 6 * 3 Models**

	SST2	SST5	MR	Subj	AGNews	CB
# Validation Examples	872	1101	1066	2000	7600	56
# Label Types	2	5	2	2	4	3
ZSL Accuracy (GPT 1.3B)	70.5	39.3	65.9	72.6	46.3	37.5
FT Accuracy (GPT 1.3B)	73.9	39.5	73.0	77.8	65.3	55.4
ICL Accuracy (GPT 1.3B)	92.7	45.0	89.0	90.0	79.2	57.1
ZSL Accuracy (GPT 2.7B)	71.4	35.9	60.9	75.2	39.8	42.9
FT Accuracy (GPT 2.7B)	76.9	39.1	80.0	86.1	65.7	57.1
ICL Accuracy (GPT 2.7B)	95.0	46.5	91.3	90.3	80.3	55.4

- **Metrics1: Rec2FTP**

$$\frac{N_{(\mathrm{FT}>\mathrm{ZSL})\wedge(\mathrm{ICL}>\mathrm{ZSL})}}{N_{\mathrm{FT}>\mathrm{ZSL}}}$$

- **Metrics1: Rec2FTP**

Model	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	91.84	66.67	97.08	87.17	83.08	87.50	85.56
GPT 2.7B	96.83	71.60	95.83	87.63	84.44	100.00	89.39

- **Result: From the perspective of model prediction, ICL can cover most of the correct behavior of finetuning**

- **Metrics2: Similarity of the attention output updates (SimAOU)**

- **SimAOU(ΔFT) =** $\frac{1}{L}\sum_{l=0}^{L-1}\cos < h_{ICL}^{l} - h_{ZSL}^{l}, h_{FT}^{l} - h_{ZSL}^{l} >$

- **SimAOU(RandomΔ) =** $\frac{1}{L}\sum_{l=0}^{L-1}\cos < h_{ICL}^{l} - h_{ZSL}^{l}, h_{\mathrm{Random}\,\Delta}^{l} >$

- h_{x}^{l} **- The normalized output representation of the last token at the l-th attention layer in setting X**

- **Metrics2: Similarity of the attention output updates (SimAOU)**

Model	Metric	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	SimAOU (Random Δ)	0.002	0.003	0.001	0.002	0.002	0.003	0.002
	SimAOU (ΔFT)	**0.110**	**0.080**	**0.222**	**0.191**	**0.281**	**0.234**	**0.186**
GPT 2.7B	SimAOU (Random Δ)	0.000	-0.002	0.000	0.001	-0.002	0.000	-0.001
	SimAOU (ΔFT)	**0.195**	**0.323**	**0.157**	**0.212**	**0.333**	**0.130**	**0.225**

Result: ICL updates are much more similar to finetuning updates than to random updates. From the perspective of representation, ICL tends to change attention output representations in the same direction as finetuning changes.

- **Metrics3: Similarity of the attention map (SimAM)**

- **SimAM(Bedore fine tuning) =** $\frac{1}{LH}\sum_{l=0}^{L-1}\sum_{l=0}^{L-1}\cos < \mathrm{m}_{ICL}^{l,h}, \mathrm{m}_{ZSL}^{l,h} >$
- **SimAM(After fine tuning) =** $\frac{1}{LH}\sum_{l=0}^{L-1}\sum_{l=0}^{L-1}\cos < \mathrm{m}_{ICL}^{l,h}, \mathrm{m}_{FT}^{l,h} >$

- h_x^l **- The attention weights before softmax of the last token at the h-th attention head in the l-th attention layer in setting X. For ICL, we omit the attention to the demonstration tokens**
- **and only monitor the attention weights to the query**
- **tokens**

- **Metrics3: Similarity of the attention map (SimAM)**

Model	Metric	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	SimAM (Before Finetuning)	0.555	0.391	0.398	0.378	0.152	0.152	0.338
	SimAM (After Finetuning)	**0.585**	**0.404**	**0.498**	**0.490**	**0.496**	**0.177**	**0.442**
GPT 2.7B	SimAM (Before Finetuning)	**0.687**	0.380	0.314	0.346	0.172	**0.228**	0.355
	SimAM (After Finetuning)	**0.687**	**0.492**	**0.347**	**0.374**	**0.485**	0.217	**0.434**

- **Result: From the perspective of attention behavior, compared with attettion weights before finetuning, ICL is more inclined to generate similar attention weights to those after finetuning**

- **Metrics4: Kendall rank correlation coefficient**

- **Kendall (ICL, FT) =** $\frac{1}{L}\sum_{l=0}^{L-1}\mathrm{Kendall}(m_{ICL}^{l}, m_{FT}^{l})$
- **Kendall (ICL, Random) =** $\frac{1}{L}\sum_{l=0}^{L-1}\mathrm{Kendall}(m_{ICL}^{l}, m_{random}^{l})$

- $m_x^l = \sum_{h=0}^{H-1} K^{l,h} * q^{l,h}$

- **The x setting attention weights to the demonstration tokens of the last query token in the l-th attention layer, which is summed across attention heads.**

- **Metrics4: Kendall rank correlation coefficient**

Model	Metric	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	Kendall (ICL, Random)	0.000	-0.001	0.000	0.001	-0.001	0.000	0.000
	Kendall (ICL, FT)	**0.192**	**0.151**	**0.173**	**0.181**	**0.190**	**0.274**	**0.193**
GPT 2.7B	Kendall (ICL, Random)	-0.001	0.000	0.000	0.000	0.000	-0.001	0.000
	Kendall (ICL, FT)	**0.213**	**0.177**	**0.264**	**0.203**	**0.201**	**0.225**	**0.214**

- **Result: Compared with random attention weights, ICL attention weights to training tokens are much more similar to finetuning attention weights.**

- **New attention Mechnism: Momentum-Based Attention**

- **Momutem Averaged Gradient Descent:**

$$\Theta_t = \Theta_{t-1} - \gamma \sum_{i=1}^{t-1} \eta^{t-i} \nabla f_{\Theta_i}$$

- **Momutem-Based Attention:**

$$\begin{aligned} \mathrm{MoAttn}(V, K, \mathbf{q}_t) &= \mathrm{Attn}(V, K, \mathbf{q}_t) + \mathrm{EMA}(V) \\ &= V\,\mathrm{softmax}\left(\frac{K^T \mathbf{q}_t}{\sqrt{d}}\right) + \sum_{i=1}^{t-1} \eta^{t-i} \mathbf{v}_i \end{aligned}$$

- **New attention Mechnism: Momentum-Based Attention**

Model	Train_{1024}	Valid_{256}	Valid_{512}	Valid_{1024}
Transformer	17.61	19.50	16.87	15.14
$\text{Transformer}_{\text{MoAttn}}$	**17.55**	**19.37**	**16.73**	**15.02**

- **Result: Momentum-based attention achieves a consistent perplexity improvement compared with the vanilla Transformer.**

Model	SST5	IMDB	MR	CB	ARC-E	PIQA	Average
Transformer	25.3	64.0	61.2	43.9	48.2	68.7	51.9
$\text{Transformer}_{\text{MoAttn}}$	**27.4**	**70.3**	**64.8**	**46.8**	**50.0**	**69.0**	**54.7**

- **Result: Introducing momentum into attention improves the accuracy of the vanilla Transformer by 2.8 on average.**

Conclusion & Contribution

- Prove ICL behaves similarly to explicit finetuning from multiple perspectives by experiments.
- Inspired by understanding of ICL as implict gradient-descent, designs a momentum-based attention that achieves consistent performance improvements over vanilla attention

Limitations

- Limited Scope to Transformer-based In-Context Learning
- Simplified Treatment of Transformer Attention and Gradient Descent Dualism
- The paper only train GPT models up to 2.7B. But didn’t research on larger model like GPT13B
- Classification Task Focus

Q&A

Thanks