南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

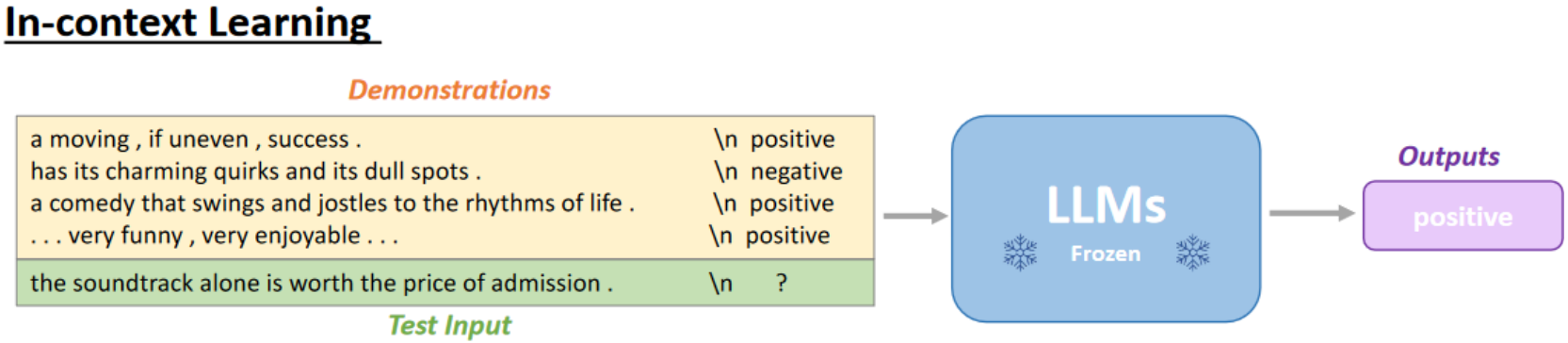
On the Noise Robustness of In-Context Learning for Text Generation

Introduction - ICL

- Given a new test input text $\mathbf{x}_{\text{test}}$, we make the generation of output $\mathbf{y}_{\text{test}}$ via large language models as

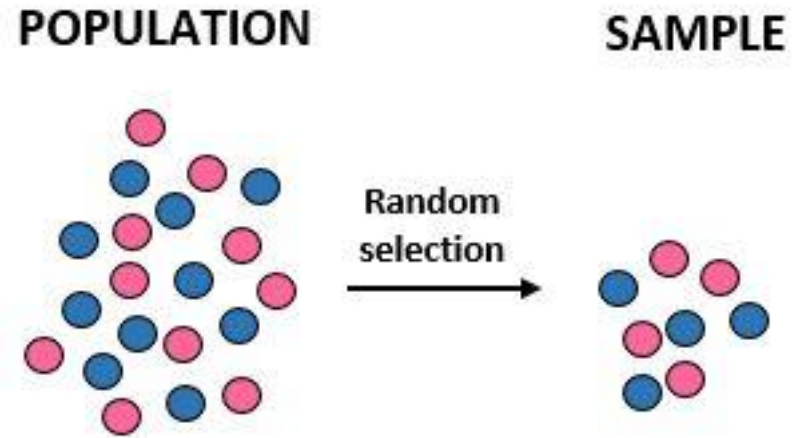
$$\mathbf{y}_{\text{test}} \sim \mathrm{P}_{\mathrm{LLM}}(\boldsymbol{y}_{\text{test}} | \{(\mathbf{x}_{\mathrm{i}}, \mathbf{y}_{\mathrm{i}})\}_{\mathrm{i}=1}^{\mathrm{K}}, \mathbf{x}_{\text{test}})$$

where the context $C_K = \{(\mathbf{x}_{\mathrm{i}}, \mathbf{y}_{\mathrm{i}})\}_{\mathrm{i}=1}^{\mathrm{K}}$ contains K task demonstrations, selected from a large annotated dataset with N examples $\mathcal{D} = \{(\mathbf{x}_{\mathrm{j}}, \mathbf{y}_{\mathrm{j}})\}_{\mathrm{j}=1}^{\mathrm{N}}$.



Demonstration Selection is Crucial

- **Random**: A naive method is to randomly sample the demonstrations from annotated data without repetition.



Demonstration Selection - TopK

- **TopK** proposes to select the closest examples to the test input in the embedding space

$$C_K = R_K(\mathbf{x}_{\text{test}}) = \text{TopK}_{\mathbf{x}}(s(\mathbf{x}_{\text{test}}, \mathbf{x}))$$

where $s(\mathbf{x}_{\text{test}}, \mathbf{x})$ denotes the cosine similarity score between $\mathbf{x}_{\text{test}}$ and $\mathbf{x}$ from the annotated dataset.

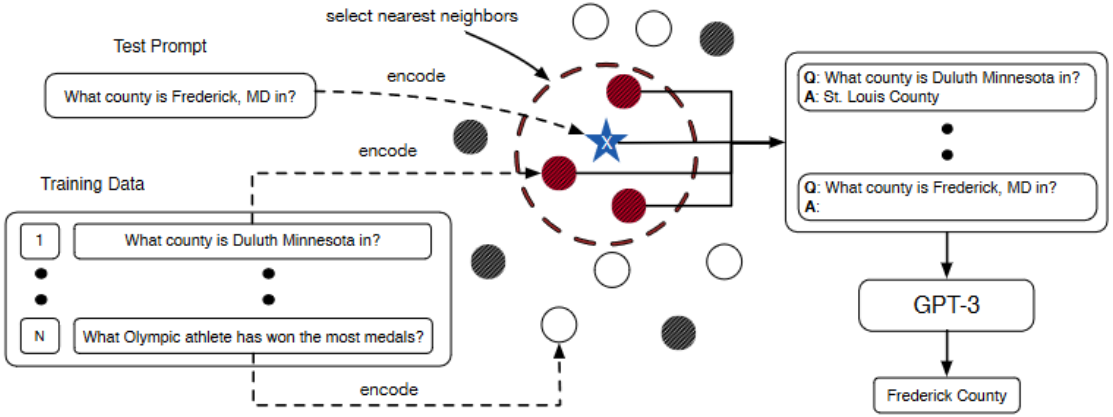
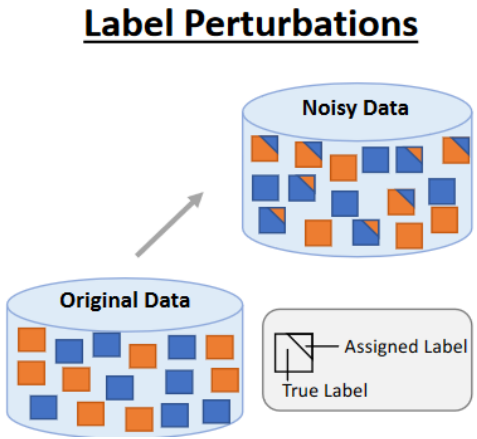


Figure 1: In-context example selection for GPT-3. White dots: unused training samples; grey dots: randomly sampled training samples; red dots: training samples selected by the k-nearest neighbors algorithm in the embedding space of a sentence encoder.

Limitation of Clean Hypothesis

- These selection strategies focus on the inputs of demonstrations, **assuming that all examples are labeled correctly in the large dataset**.
- In practice, researchers often use crowdsourcing or large language models (LLMs) such as GPT-4 to create input-output pairs for new tasks, which **inevitably leads to some mistakes in the annotations**.



ICL with noisy demonstrations

- Conditioned on the noisy demonstrations, the generation of output via ICL is made as

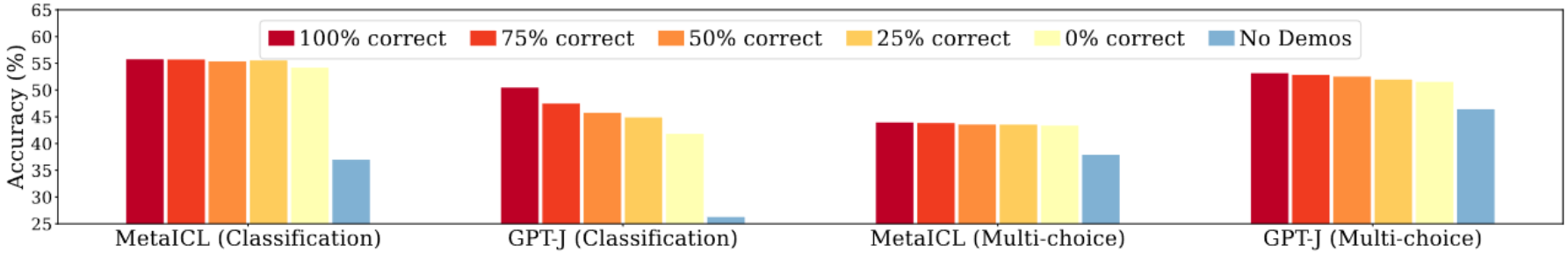
$$\mathbf{y}_{\text{test}} \sim \mathrm{P}_{\text{LLM}}\left(\boldsymbol{y}_{\text{test}} \middle| \{(\mathbf{x}_{\mathrm{i}}, \tilde{\boldsymbol{y}}_{\mathrm{i}})\}_{\mathrm{i}=1}^{\mathrm{K}}, \mathbf{x}_{\text{test}}\right)$$

where $\{(\mathbf{x}_{\mathrm{i}}, \tilde{\boldsymbol{y}}_{\mathrm{i}})\}_{\mathrm{i}=1}^{\mathrm{K}}$ are selected from a large-scale dataset with noisy annotations $\mathcal{D} = \{(\mathbf{x}_{\mathrm{j}}, \tilde{\boldsymbol{y}}_{\mathrm{j}})\}_{\mathrm{j}=1}^{\mathrm{N}}$,

and the output $\tilde{\mathbf{y}}$ might be **not** a correct answer to the input $\mathbf{x}$.

Mainstream View: label noises do not harm ICL

Previous works (Min et al., 2022; Fei, et al., 2023; Lyu et al., 2023) show that **in-context learning on classification tasks is fairly robust to label noise in the in-context demonstrations**.



Results with varying number of correct labels in the demonstrations in tasks (Min et al., 2022). The results are evaluated on 16 classification and 10 multi-choice datasets.

Limitation:

1. It is still mysterious how noisy labels affect the performance of ICL on **text generation tasks**.
2. The existing studies only focus on **Random** demonstration selection method.

Empirical study of noisy ICL in text generation

We define two categories of noisy annotations based on the input-output relevance.

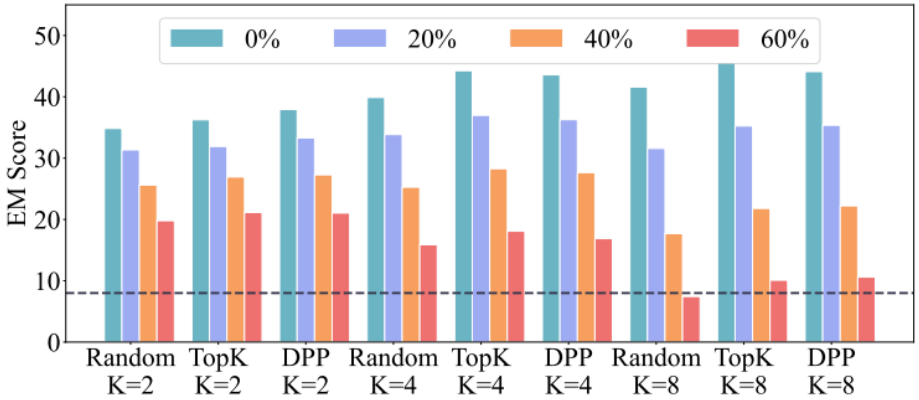
1. **Irrelevant noise** assumes that the generation of noisy annotations is conditionally independent of inputs.
2. **Relevant noise** is a more realistic setting where the corrupted output is relevant to the inputs despite its incorrectness.

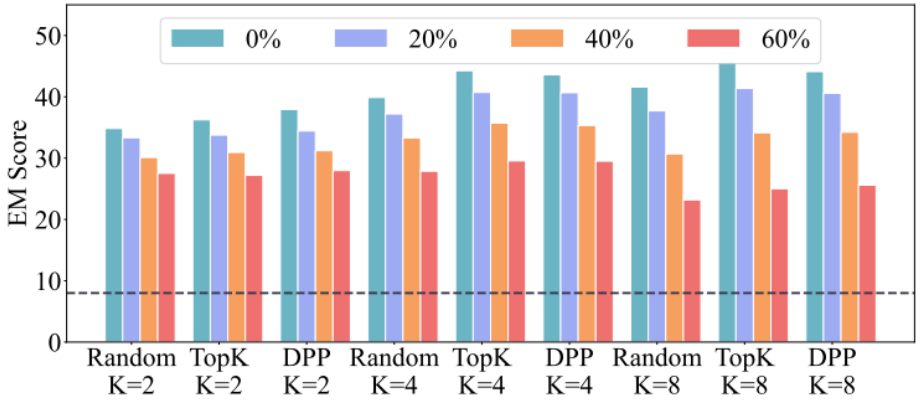
Test Input	**Support**: All forms of life are built of at least one cell. A cell is the basic unit of the structure and function of living things. **Question**: What are the smallest structural and functional units of all living organisms? **Output**:	
Setting	**In-Context Demonstration**	**Prediction**
Clean	**Support**: Cells are organized into tissues, tissues are organized into organs. **Question**: What is considered the smallest unit of the organ? **Output**: Cells	Cells
Irrelevant	**Support**: Cells are organized into tissues, tissues are organized into organs. **Question**: What is considered the smallest unit of the organ? **Output**: Earth	Earth
Relevant	**Support**: Cells are organized into tissues, tissues are organized into organs. **Question**: What is considered the smallest unit of the organ? **Output**: tissues	tissues

Empirical findings of noisy ICL in text generation

Text generation tasks: **question answering, reading comprehension, code generation**.

- ICL is **not robust** to noisy annotations in text generation.
- Selecting a larger set of demonstrations even worsen the performance of text generation.
- The advantages of those powerful selection methods (i.e., TopK and DPP) are neutralized.





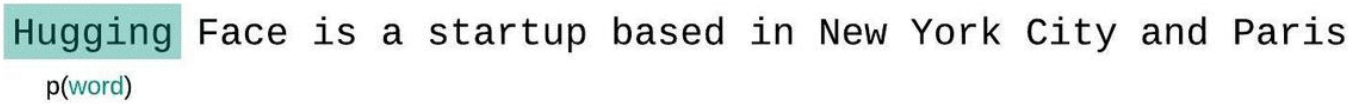
Perplexity in LLM

- The **perplexity** of tokenized input-output pair $\mathbf{z}$ is calculated as:

$$\text{Perplexity}(\mathbf{z}) = \exp\{-\frac{1}{|\mathbf{z}|}\sum_{i=1}^{|\mathbf{z}|}\log p_\theta(\mathbf{z}_i|\mathbf{z}_{<i})\}$$

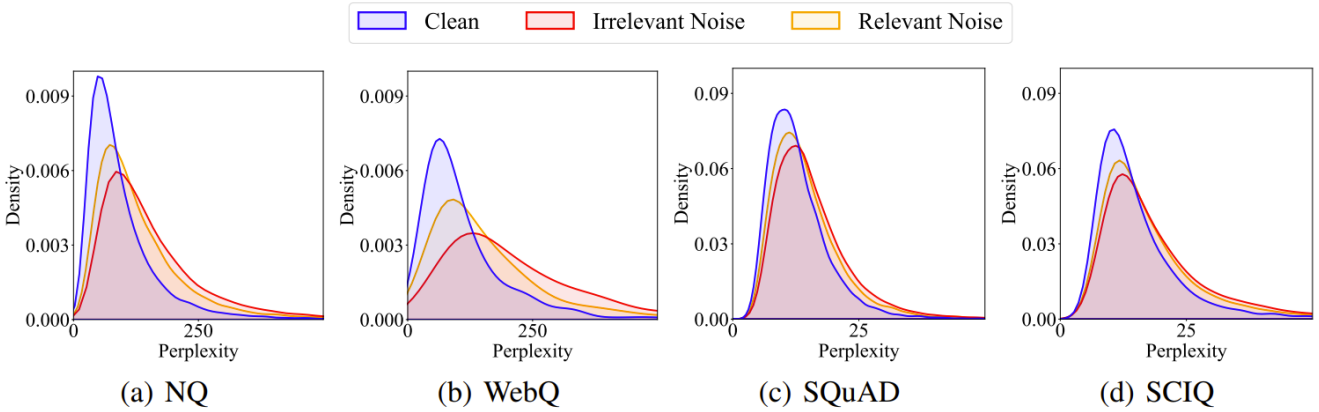
where $\log p_\theta(\mathbf{z}_i|\mathbf{z}_{<i})$ is the log-likelihood of the i-th token conditioned on the preceding tokens $\mathbf{z}_{<i}$ from the given language model parameterized by θ.

- For language models, perplexity measures **the degree of uncertainty** in generating new tokens. A **low perplexity** indicates that the model makes the prediction with **high confidence**.



Perplexity deviation of noisy annotations

- Examples with noisy annotations indeed obtain **higher perplexity** than those with clean annotations,
- **Relevant noises achieve slightly lower perplexity than irrelevant noises** since relevant outputs are close to the inputs despite their erroneous information.
- However, the deviation of the perplexity distribution caused by noisy annotations is marginal, **making it challenging to differentiate noisy annotations from clean ones**.



Disentanglement of perplexity

Informally, we decompose the overall Perplexity into two components, as shown below:

$$\text{Perplexity} = \text{Inherent Perplexity} + \text{Matching Perplexity}$$

1. **Inherent perplexity**

   how the model is familiar with the task (i.e., the input and the correct output).

2. **Matching perplexity**

   the perplexity deviation caused by noisy outputs, so it can be zero with correct outputs.

Question: How to compare the Matching Perplexity of demonstrations?

Two assumptions of our method

Here, our approach is built on two natural assumptions that are naturally satisfied in the real world:

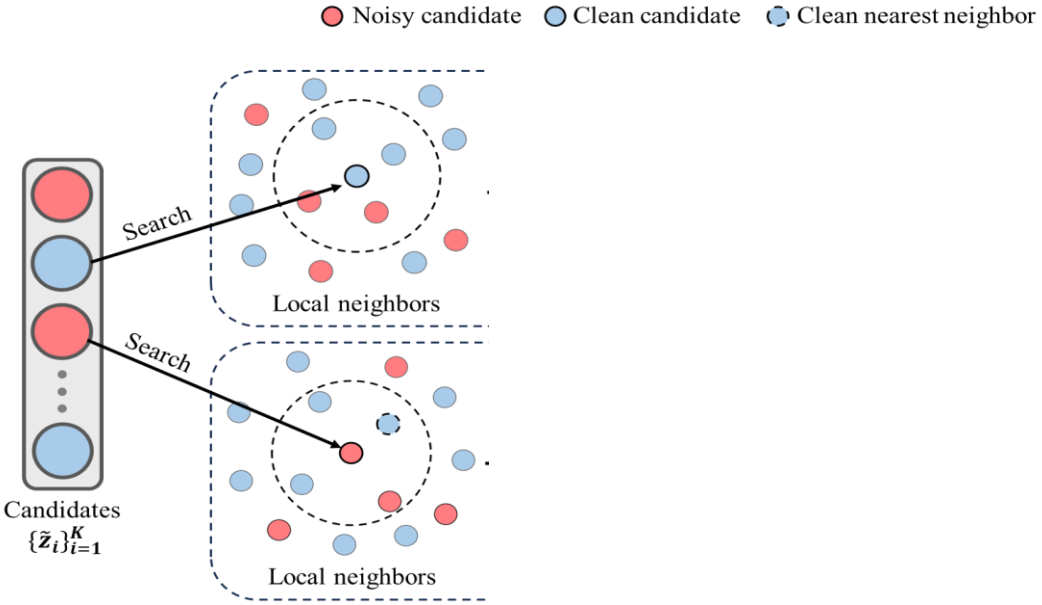
1. **The clean annotations are the majority in the annotated dataset**.

2. **Examples that are semantically similar share the same level of inherent perplexity**.

The candidate is more likely to be wrongly annotated

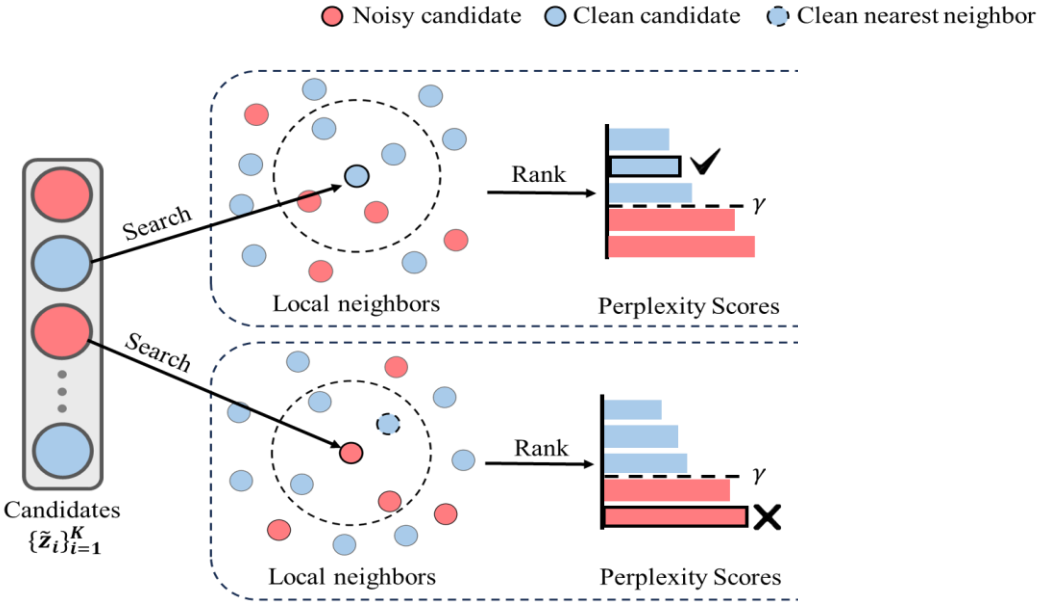
if its perplexity is relatively higher than its neighbors

Local Perplexity Ranking



- **Finding the local neighbors**: For each candidate $\mathbf{z}^*$, we adopt k-Nearest-Neighbors (k-NN) to find its local neighbors $\mathbf{z}_n$ that are close to the candidate in token space.

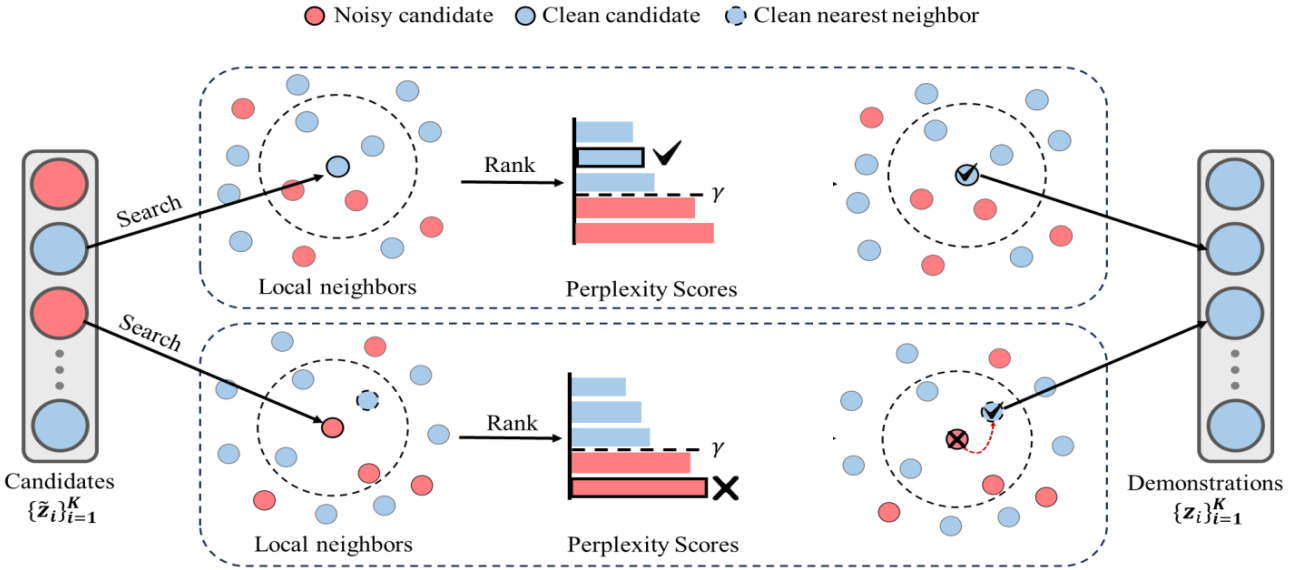
Local Perplexity Ranking



- **Ranking the perplexity:** For each candidate $\mathbf{z}^*$, we sort all examples in the cluster in increasing order by the perplexity and obtain the original indices for the sorted scores as:

$$I = \text{argsort}\{\text{Perplexity}(\mathbf{z}_\text{n})\}_{n=1}^{k+1}, \mathbf{z}_\text{n} \in (\mathbf{z}^* \cup N_k(\mathbf{z}^*))$$

Local Perplexity Ranking



- **Substituting the noisy candidates:** We determine whether a candidate should be replaced by:

$$g(\mathbf{z}_n) = \mathbb{I}\left(\frac{\mathrm{Loc}(\mathbf{z}_n, \mathrm{I})}{k+1} \geq \gamma\right)$$

where γ is the pre-defined threshold, $\mathbb{I}(\cdot)$ is the indicator function and $\mathrm{Loc}(\mathbf{z}_n, \mathrm{I})$ return the index of $\mathbf{z}_n$ in the sorted list I.

Advantages of LPR

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Algorithm-agnostic

LPR can be easily incorporated into existing demonstration selection methods, consistently improving the robustness against noisy annotations.

Easy to use

LPR does not require heavy hyperparameter tuning, as it is insensitive to the threshold value. LPR does not introduce much computational cost due to the efficient computation of perplexity.

Experiments

- We employ 6 generation datasets for the evaluations, including open-domain question answering (NQ, WebQ), reading comprehension (SQuAD, SCIQ), code generation (GeoQuery, NL2Bash).

- Our method drastically improves the noise-robustness performance of the existing ICL demonstration selection methods on 6 generation datasets.

Dataset	Method	Clean	Irelevant Noise			Relevant Noise		
		0%	20%	40%	60%	20%	40%	60%
NQ	Random	14.51±0.51	10.97±0.29	7.37±0.45	4.23±0.46	12.00±0.65	9.67±0.45	6.40±1.02
	+Ours	**15.05±0.10**	**13.31±0.25**	**11.51±0.51**	**8.87±0.74**	**13.74±0.12**	**13.28±0.33**	**9.43±0.52**
	TopK	20.25±0.10	13.95±1.14	9.97±1.13	5.90±1.08	16.21±0.22	12.22±0.22	8.50±0.28
	+Ours	19.19±0.19	**17.15±0.50**	**13.54±0.41**	**9.64±0.25**	**17.25±0.69**	**14.82±0.51**	**11.98±0.60**
	DPP	20.35±0.76	14.69±0.94	9.87±0.49	5.97±0.48	15.47±1.00	11.28±0.42	7.89±0.25
	+Ours	19.68±0.33	**16.59±0.45**	**13.31±0.57**	**11.18±0.50**	**16.79±0.47**	**14.91±0.18**	**11.94±0.91**
WebQ	Random	20.37±0.64	15.18±1.06	10.39±0.83	4.83±0.17	18.29±0.43	15.92±0.68	13.50±0.17
	+Ours	**21.94±0.64**	**20.32±0.92**	**16.33±0.58**	**12.54±0.29**	**21.51±0.33**	**19.33±0.41**	**16.69±1.11**
	TopK	30.16±0.58	22.52±0.64	14.52±0.78	8.00±1.12	27.19±0.27	22.82±0.75	18.88±1.09
	+Ours	29.24±0.34	**26.55±0.24**	**21.67±1.28**	**14.54±1.02**	**28.49±0.43**	**25.44±0.68**	**21.28±0.12**
	DPP	29.40±0.39	22.11±0.81	13.72±0.27	7.33±0.68	26.18±1.04	21.53±0.61	16.80±0.17
	+Ours	**29.92±0.48**	**26.57±0.95**	**21.94±1.05**	**14.85±0.81**	**28.46±1.01**	**25.61±0.78**	**21.35±1.17**
SQuAD	Random	56.50±0.57	50.00±0.62	39.10±0.88	26.20±0.79	53.90±0.65	49.17±0.62	42.03±0.79
	+Ours	**57.73±0.79**	**56.87±0.47**	**48.50±0.86**	**43.00±0.86**	**57.70±1.31**	**53.93±0.33**	**47.93±0.48**
	TopK	56.97±0.69	51.83±1.03	42.83±1.68	29.10±2.92	54.77±0.69	49.37±1.37	41.37±2.09
	+Ours	**57.27±0.62**	**55.40±0.37**	**51.43±1.26**	**41.30±2.65**	**56.90±0.64**	**53.90±1.08**	**48.37±0.66**
	DPP	57.29±0.87	50.57±0.33	41.63±1.00	25.67±2.52	56.10±0.59	49.57±1.24	43.37±0.78
	+Ours	**58.10±0.29**	**56.73±0.61**	**52.53±0.33**	**42.93±0.88**	**57.50±0.54**	**55.90±0.18**	**50.77±0.39**
SCIQ	Random	68.15±0.28	59.19±1.57	44.19±2.89	28.21±2.96	64.59±1.42	58.39±0.16	49.54±0.80
	+Ours	67.93±0.85	**65.06±1.34**	**55.57±0.53**	**42.00±2.96**	**66.63±0.94**	**62.70±1.10**	**58.92±1.74**
	TopK	68.62±1.13	59.59±1.28	45.77±2.68	29.31±1.73	64.66±1.34	58.54±0.12	49.47±0.65
	+Ours	**70.06±0.32**	**66.67±0.81**	**57.44±1.04**	**48.06±1.53**	**67.76±0.50**	**63.96±1.71**	**56.32±2.18**
	DPP	67.29±0.35	57.69±1.83	45.34±1.56	28.50±1.78	64.88±0.43	58.91±0.64	50.00±0.85
	+Ours	**70.57±0.45**	**67.86±1.43**	**59.65±2.11**	**45.46±2.72**	**69.16±0.98**	**65.63±0.21**	**56.72±1.37**
GeoQuery	Random	27.97±0.99	23.18±0.62	17.44±1.56	14.10±0.74	26.48±0.17	26.13±0.05	26.25±0.40
	+Ours	27.27±0.36	**27.12±0.69**	**25.52±1.02**	**22.23±0.67**	**27.43±0.71**	**27.01±0.05**	**26.73±0.90**
	TopK	44.17±0.09	27.28±2.65	17.49±2.05	9.96±3.08	41.31±0.46	38.48±0.63	34.90±0.69
	+Ours	43.32±0.05	**42.25±1.00**	**33.80±1.43**	**24.39±1.08**	**42.59±0.37**	**39.40±0.37**	**37.74±1.23**
	DPP	45.81±0.71	31.79±5.93	21.54±3.36	10.61±0.15	42.97±1.96	39.91±0.42	33.34±0.53
	+Ours	44.18±0.47	**43.01±0.02**	**40.94±0.91**	**33.25±1.27**	41.49±0.11	**40.62±0.06**	**36.81±0.61**
NL2Bash	Random	27.91±0.37	25.37±0.21	15.77±0.91	8.95±0.65	27.20±1.06	28.09±0.51	26.27±0.56
	+Ours	**29.93±1.18**	**29.09±0.26**	**26.04±2.05**	**22.92±0.39**	**29.01±0.36**	**28.92±0.07**	**26.80±0.55**
	TopK	35.71±0.42	27.40±0.26	20.00±0.62	9.95±0.68	32.57±0.13	30.21±0.08	27.48±0.35
	+Ours	33.92±0.70	**32.51±1.59**	**30.50±1.02**	**23.47±1.52**	31.33±0.04	**31.39±1.70**	**29.49±0.06**
	DPP	37.77±0.02	31.52±0.12	23.23±0.34	11.16±2.14	32.74±0.29	32.56±0.61	26.72±1.58
	+Ours	35.85±1.51	**32.27±0.99**	**32.47±0.40**	**27.84±1.17**	**33.63±0.23**	32.53±0.57	**28.96±0.98**

11.16±2.14
27.84±1.17

The performance of LPR is **insensitive** to the hyperparameters

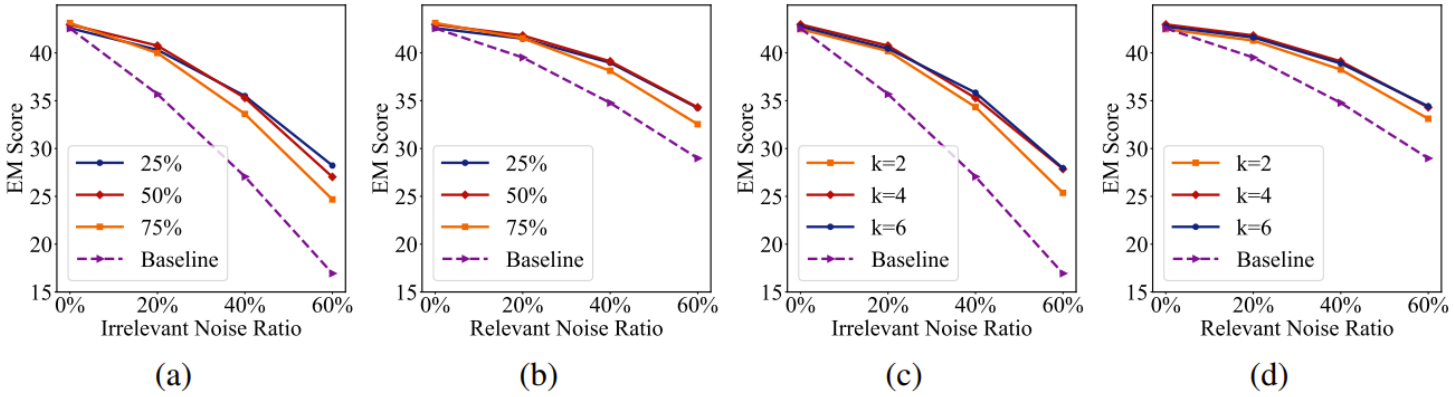


Figure 3: The average test performance with different thresholds τ and numbers of local neighbors k across various noise types. The performance of LPR is insensitive to the hyperparameters.

Experiments

The larger the LLM is, the higher the improvement we get.

Table 3: Average test performance using varying large language models across various noise types. The results are shown as Naive/+Ours. The bold indicates the improved results by integrating LPR.

Method	Clean	Irelevant Noise			Relevant Noise		
	0%	20%	40%	60%	20%	40%	60%
Llama2-13B [47]	45.13/**45.27**	38.58/**43.47**	29.00/**39.24**	18.93/**30.46**	42.18/**44.32**	37.10/**41.88**	30.67/**36.76**
Mistral-7B [19]	34.89/34.12	32.12/**33.59**	26.28/**31.56**	19.24/**27.03**	33.43/**33.91**	30.52/**32.64**	26.63/**30.00**
OPT-6.7B [63]	23.46/**24.03**	17.26/**21.31**	11.32/**17.29**	7.68/**12.91**	20.16/**22.40**	17.58/**20.22**	14.95/**17.52**

Experiments

LPR also works for **text classification tasks**.

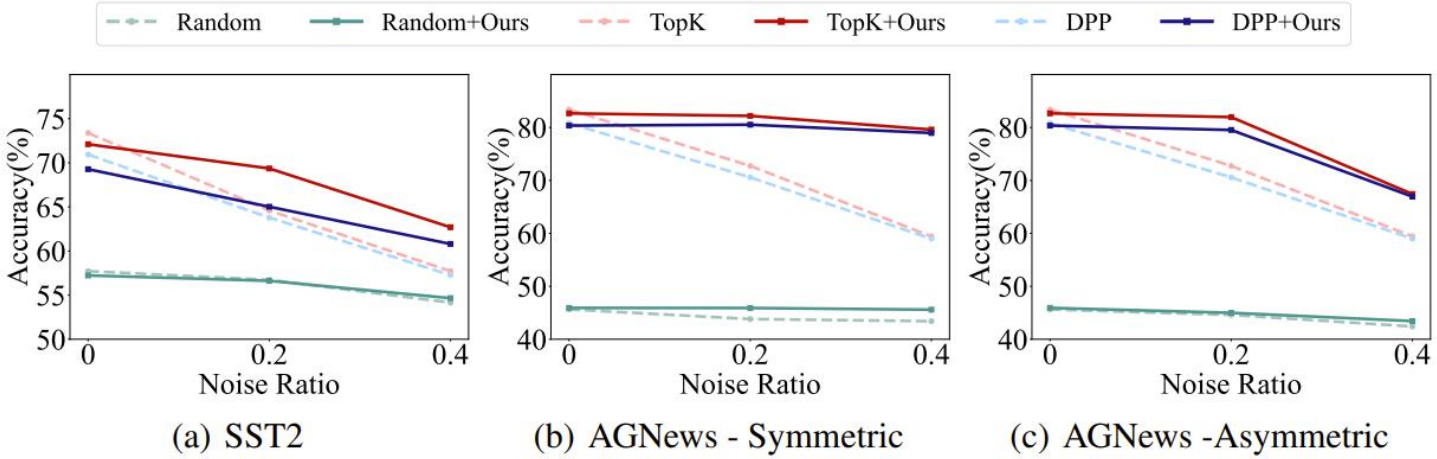


Figure 4: Average test accuracy on SST2 [46] and AGNews [64]. Different colors indicate the selection methods. The solid lines denote existing selection methods, and the dotted lines represent the method integrated by our method. We omit the noisy type on the binary classification – SST2.

Discussion

One may also ask: *can a similar effect be achieved by selecting demonstrations with the lowest perplexity in the whole dataset?*

- The global approach obtains inferior performance compared to our proposed method in most cases, especially in the cases of clean and low noise rates.
- The local ranking approach requires only 20% of the time required by the global ranking.

Table 4: Average test performance comparison between global perplexity ranking and local perplexity ranking. The results are shown as *Global/Local*. Bold numbers are superior results.

Method	Clean	Irrelevant Noise			Relevant Noise			Time (h)
	0%	20%	40%	60%	20%	40%	60%	
Random	39.32/**40.66**	38.94/38.89	**34.41**/32.98	**27.82**/26.59	39.23/**39.90**	36.38/**37.31**	31.76/**33.24**	3.48/**0.55**
TopK	40.57/**43.94**	39.94/**41.44**	35.85/**36.02**	**31.79**/28.38	40.33/**42.60**	38.69/**39.53**	33.88/**34.48**	3.59/**0.57**
DPP	42.33/**44.32**	40.18/**41.94**	36.20/**36.86**	**30.91**/28.60	40.42/**42.98**	38.49/**40.51**	32.24/**35.20**	3.97/**0.64**
Average	40.74/**42.97**	39.68/**40.76**	**35.49**/35.28	**30.17**/27.86	39.99/**41.83**	37.85/**39.12**	32.63/**34.31**	3.68/**0.59**

Thanks!