Bayesian Example Selection Improves In-Context Learning for Speech, Text and Visual Modalities

Siyin Wang[1], **Chao-Han Huck Yang**[2], **Ji Wu**[1], **Chao Zhang**[1]*
[1]Tsinghua University, [2]NVIDIA Research
wangsiyi23@mails.tsinghua.edu.cn,cz277@tsinghua.edu.cn

Abstract

Large language models (LLMs) can adapt to new tasks through in-context learning (ICL) based on a few examples presented in dialogue history without any model parameter update. Despite such convenience, the performance of ICL heavily depends on the quality of the in-context examples presented, which makes the in-context example selection approach a critical choice. This paper proposes a novel **Ba**yesian in-**C**ontext example **S**election method (ByCS) for ICL. Extending the inference probability conditioned on in-context examples based on Bayes' theorem, ByCS focuses on the inverse inference conditioned on test input. Following the assumption that accurate inverse inference probability (likelihood) will result in accurate inference probability (posterior), in-context examples are selected based on their inverse inference results. Diverse and extensive cross-tasking and cross-modality experiments are performed with speech, text, and image examples. Experimental results show the efficacy and robustness of our ByCS method on various models, tasks and modalities.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023b; OpenAI, 2023a) have achieved great success on many text-based natural language processing (NLP) tasks. By connecting with extra visual and audio encoders (Sun et al., 2023b; Radford et al., 2023), the resulting multimodal LLMs can also achieve remarkable performance on image-text and audio-text tasks (Li et al., 2023; OpenAI, 2023b; Tang et al., 2023). With the ability of in-context learning (ICL) (Brown et al., 2020), LLMs can adapt to new tasks easily and efficiently in a training-free manner, to generate output following the prompting paradigm based on a few input-label pairs pre-pended to the test input. The existence of ICL ability has also been verified on image-text and audio-text tasks (Tsimpoukelli et al., 2021; Wang et al., 2023c; Hsu et al., 2023; Pan et al., 2023).

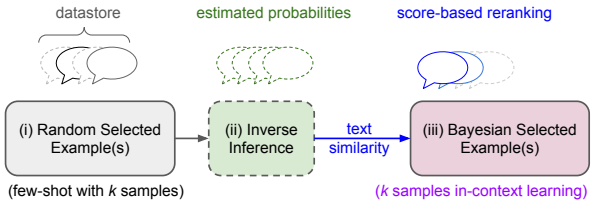


Figure 1: A brief illustration of the proposed Bayesian in-context example selection includes: (i) first randomly selecting k examples; (ii) examining the examples in the datastore through "inverse inference," where the test input-label pair serves as the in-context example; and (iii) selecting samples with correct label predictions as good examples (colored in blue), considered to have high mutual information interaction with the test input.

Although ICL requires no gradient descent and thus does not suffer from the instability caused by stochastic optimisation compared to other test-time adaptation approaches, care still needs to be taken when selecting the in-context examples since they often lead to distinct ICL performance variations (Zhao et al., 2021; Min et al., 2022; Lu et al., 2022b). Prior work on in-context example selection trains an example retrieval module (Rubin et al., 2022; Zhang et al., 2022; Lu et al., 2022a; Wang et al., 2023b), selects close examples in embedding space (Liu et al., 2022; An et al., 2023; Qin et al., 2023), or leverages the feedback of LLMs to score the examples (Su et al., 2022; Nguyen and Wong, 2023; Iter et al., 2023; Mavromatis et al., 2023). While boosting ICL performance, most methods treat in-context examples and test input separately, overlooking their mutual interactions.

This paper proposes ByCS (**Ba**yesian in-**C**ontext example **S**election), a novel in-context example selection approach focusing on mutual information interactions based on the Bayesian formula. Refer to the inference of test input conditioned on in-context examples as ICL *inference*, and the

*Corresponding author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 20812–20828
November 12-16, 2024

inference of in-context example's input based on the test input-label pair as the *inverse inference*. By introducing inverse inference via *Bayes' theorem*, ByCS leverages the inverse inference result to evaluate the quality of each in-context example. Assuming the contextual information interaction is mutual, an accurate inverse inference is likely to result in an accurate inference. Examples with accurate inverse inference results are selected as optimal examples. Extensive experiments across audio, image, and text modalities are conducted to verify the effectiveness and robustness of ByCS, such as ASR, visual question answering (VQA), as well as NLP tasks (including topic classification, sentiment analysis, and text-to-SQL *etc*). Our main contributions are summarised as follows:

- ByCS, a novel in-context example selection method inspired by Bayes' theorem, is proposed. To improve the efficiency, the use of a smaller model for fast inverse inference implementation and a ranking-based pre-selection to reduce the number of in-context examples are also proposed in this paper.
- The method is verified using both "decoder-only ICL" on NLP tasks and "encoder-decoder" ICL on ASR and VQA. To the best of our knowledge, this is the first work of an in-context example selection method verified across text, audio, and visual modalities as shown in Figure 2.

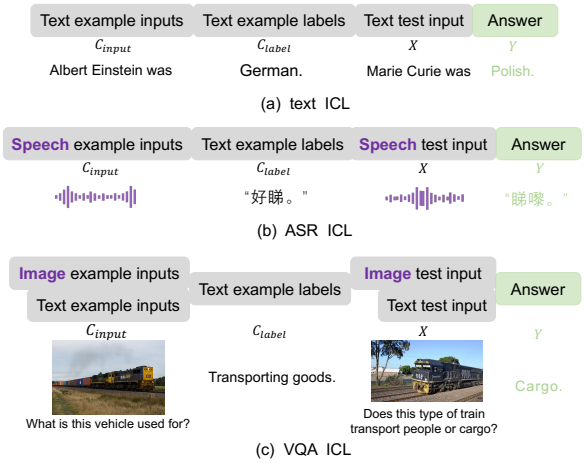


Figure 2: Multimodal ICL. Although ICL on different modalities shares the same formula expression, the actual inputs and inference model architectures differ. For ASR ICL on Whisper, the speech is fed into the encoder while the text example is labelled into the decoder, which is aware of speech input through cross-attention with the encoder. For VQA ICL, images are first encoded to the same embedding space of LM's input, then interleaved images and texts are fed into decoder LM.

2 Related Work

Multimodal ICL. Inspired by the decoder-only ICL in text-based NLP, efforts have been made to extend such a few-shot learning ability to other modalities, in particular image and audio. Frozen (Tsimpoukelli et al., 2021) is the first attempt to exploit ICL ability in the vision-language model (VLM). By using a vision encoder to map the input image to textual tokens in the input embedding space of a frozen text language model, Frozen can handle interleaved image and text input and achieve image-text ICL. Other work manages to improve VLM's ICL ability by using adapter blocks (Eichenberg et al., 2022), adding blockwise modality fusion structures (Alayrac et al., 2022) and scaling up the model size (Sun et al., 2023a).

In audio modality, Borsos et al. (2023) proposed AudioLM, a language model based on quantised audio tokens for audio generation tasks, which exhibits ICL ability for audio continuation. Similarly, Wang et al. (2023a) proposed VALL-E, a controllable text-to-speech synthesis system with ICL ability based on audio and text prompts. Wang et al. (2023c) presented the first ICL work for ASR based on paired speech-text examples, which adapted the Whisper (Radford et al., 2023) model to receive considerable word error rate (WER) reductions on unseen Chinese dialects. Further explorations enabled the recent speech-language models to perform ICL on more speech input tasks through warmup training (Hsu et al., 2023) or speech instruction-tuning (Pan et al., 2023).

In-Context Example Selection Methods. Rubin et al. (2022) proposed a scoring LM to retrieve in-context examples using contrastive learning, which can also be trained with reinforced learning algorithms, such as Q-learning (Zhang et al., 2022) and policy gradient (Lu et al., 2022a). Alternatively, examples that are semantically similar to the test input can be selected. Liu et al. (2022) proposed to select the k nearest neighbours (kNN) in the embedding space of the examples. When combining with chain-of-thought (Wei et al., 2022), Qin et al. (2023) proposed to select examples in the embedding space of the reasoning path. LLM feedback is often used in in-context example selection.

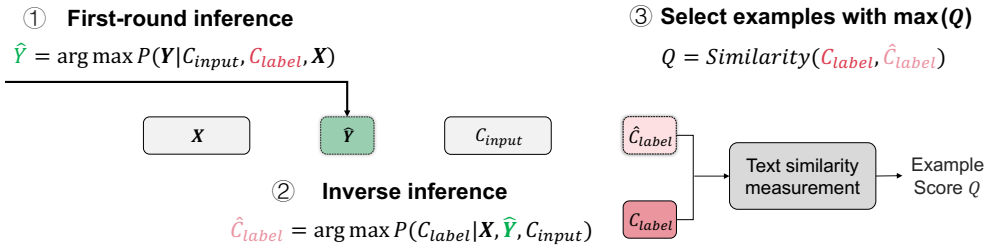


Figure 3: The detailed pipeline of our ByCS method includes: First, conduct the first-round inference to estimate the label of the test input. Then, perform inverse inference on each example in the datastore, where the test input and the estimated label serve as in-context examples. A detailed illustration of inverse inference can be found in Figure 5 in the Appendix. Finally, rank in-context examples by the text similarity between the inverse inference result and the true context label. Examples with high similarity scores are selected due to their high mutual information interaction.

Iter et al. (2023) selected in-context examples with cross-entropy differences of the fine-tuned model based on the assumption that ICL may act as implicit gradient descent (Dai et al., 2022). Nguyen and Wong (2023) identified highly impactful examples according to the proposed influence score. Although ByCS also uses LLM feedback when evaluating the quality of in-context examples through inverse inference, it leverages the text-similarity of the inverse inference results and the corresponding ground-truth labels, in no need of complete output probability distributions which are often not available for commercial LLMs.

Wang et al. (2023d) selected optimal in-context examples in the Bayesian framework by viewing LLMs as latent variable models and ICL as latent concept learning. In comparison, ByCS directly extends the ICL inference probability using Bayes' theorem. Xu and Zhang (2024) selected examples with high discrepancy between the labels and LLM's outputs when performing question answering. ByCS also selected examples from candidates in a datastore based on LLM's outputs but computes the mutual information interactions between the in-context examples and test input.

3 Methodology

As shown in Figure 3, given a test input $\mathbf{X}$ and paired in-context examples $(\mathcal{C}_{\text{input}}, \mathcal{C}_{\text{label}})$, LLMs predict the most possible answer $\hat{\mathbf{Y}}$ by maximising the inference probability $P(\mathbf{Y}|\mathcal{C}_{\text{input}}, \mathcal{C}_{\text{label}}, \mathbf{X})$:

$$\hat{\mathbf{Y}} = \arg\max P(\mathbf{Y}|\mathcal{C}_{\text{input}}, \mathcal{C}_{\text{label}}, \mathbf{X}), \quad (1)$$

where $\mathcal{C}_{\text{input}}$ and $\mathcal{C}_{\text{label}}$ are the inputs and labels of different data types in different tasks. Regarding text-based NLP tasks, $\mathcal{C}_{\text{input}}$ and $\mathcal{C}_{\text{label}}$ are referred to as text questions and corresponding answers. Regarding ASR, $\mathcal{C}_{\text{input}}$ and $\mathcal{C}_{\text{label}}$ are speech audio and corresponding text transcriptions. Regarding VQA, $\mathcal{C}_{\text{input}}$ are images and text questions based on the images and $\mathcal{C}_{\text{label}}$ are the text answers.

The inference probability can be extended using Bayes' theorem:

$$P(\mathbf{Y}|\mathcal{C}_{\text{input}}, \mathcal{C}_{\text{label}}, \mathbf{X}) = \frac{P(\mathcal{C}_{\text{label}}|\mathbf{X}, \mathbf{Y}, \mathcal{C}_{\text{input}})P(\mathbf{Y}|\mathbf{X}, \mathcal{C}_{\text{input}})}{P(\mathcal{C}_{\text{label}}|\mathbf{X}, \mathcal{C}_{\text{input}})}. \quad (2)$$

The likelihood $P(\mathcal{C}_{\text{label}}|\mathbf{X}, \mathbf{Y}, \mathcal{C}_{\text{input}})$ is termed as *inverse inference probability*, since it can be interpreted as the probability of the context label $\mathcal{C}_{\text{label}}$ when the test input-label pair $(\mathbf{X}, \mathbf{Y})$ is inversely treated as the in-context example. ByCS is focused on the inverse inference probability and assumes the influence of the prior $P(\mathbf{Y}|\mathbf{X}, \mathcal{C}_{\text{input}})$ is subordinate for simplification.

In practice, since the ground-truth label $\mathbf{Y}_{\text{ref}}$ of the test input $\mathbf{X}$ is not available, the correct likelihood $P(\mathcal{C}_{\text{label}}|\mathbf{X}, \mathbf{Y}_{\text{ref}}, \mathcal{C}_{\text{input}})$ is approximated by $P(\mathcal{C}_{\text{label}}|\mathbf{X}, \hat{\mathbf{Y}}, \mathcal{C}_{\text{input}})$, where $\hat{\mathbf{Y}}$ is produced by the first-round inference. Specifically,

- First, the first-round inference is performed to produce a hypothesized label $\hat{\mathbf{Y}}$ based on the test input $\mathbf{X}$, which can be achieved using decoding rule without any in-context examples by $\hat{\mathbf{Y}} = \arg\max P(\mathbf{Y}|\mathbf{X})$. Better performance can be achieved when using the hypothesized label obtained by in-context examples by $\hat{\mathbf{Y}} = \arg\max P(\mathbf{Y}|\tilde{\mathcal{C}}_{\text{input}}, \tilde{\mathcal{C}}_{\text{label}}, \mathbf{X})$ based on Eqn. (1), where $(\tilde{\mathcal{C}}_{\text{input}}, \tilde{\mathcal{C}}_{\text{label}})$ is a pair of first-round in-context example selected either randomly or using other example selection methods.

- Next, for the datastore with all candidate in-context examples, generate the inverse inference result in $\hat{\mathcal{C}}_{\text{label}}$ for every candidate example based on the approximated inverse inference probability $P(\mathcal{C}_{\text{label}}|\mathbf{X}, \hat{\mathbf{Y}}, \mathcal{C}_{\text{input}})$ by $\hat{\mathcal{C}}_{\text{label}} = \arg\max P(\mathcal{C}_{\text{label}}|\mathbf{X}, \hat{\mathbf{Y}}, \mathcal{C}_{\text{input}})$.

- Last, compute $\mathcal{Q} = \text{Similarity}(\mathcal{C}_{\text{label}}, \hat{\mathcal{C}}_{\text{label}})$ as the text similarity between $\mathcal{C}_{\text{label}}$ and $\hat{\mathcal{C}}_{\text{label}}$, and use $\mathcal{Q}$ as the metric for the evaluation of the quality of inverse inference. Since more accurate inverse inference probability often results in higher text similarity, ByCS selects the in-context examples with higher $\mathcal{Q}$. Note that $\mathcal{Q}$ is adopted since it does not require to assessment of the model's output probability distribution of the LLM, which is often unavailable for commercial LLMs.

To reduce the computation cost of inverse inference, two methods are used when the number of examples in the datastore is large:

- Conduct inverse inference using a model in the same model family as our inference model but has a smaller model size.

- Apply ByCS to a small number (*e.g.* N) of pre-selected candidate examples. In pre-selection, all examples in the datastore are first ranked, and only the top N best examples are reserved as the pre-selected candidates. The pre-selection is performed using fast ranking-based algorithms like kNN.

4 Experimental Setup

4.1 Models

Experimental results are performed on audio, text, and image modalities. For audio-text and image-text tasks, ASR and VQA are used to evaluate the ICL ability of encoder-decoder structured models. For text-only NLP tasks, topic classification, sentiment analysis, and text-to-SQL are used to evaluate the ICL performance with decoder-only models. Regarding the NLP tasks, experiments are conducted using GPT-3.5-Turbo and GPT-4 (OpenAI, 2023a). For the ASR task, the open-sourced Whisper model (Radford et al., 2023) is used, which is a series of speech models released by OpenAI. The Whisper model family uses vanilla encoder-decoder Transformer (Vaswani et al., 2017) architecture ranging from 39 million (M) parameters (tiny) to 1.55 billion (B) parameters (large). Specifically, the Whisper small (244M) and Whisper large-v2/-v3 (1.55B) models are used. For the VQA task, experiments are performed on Emu2 (Sun et al., 2023a) and GPT-4V (OpenAI, 2023b). Emu2 is a 37B text-image model (VLM) which leverages pre-trained EVA-02-CLIP-E-plus (Sun et al., 2023b) and LLAMA-33B (Touvron et al., 2023a), which has ICL ability when taking interleaved inputs of images and texts. For experiments on Emu2, the outputs are generated using a greedy decoding setting for fast evaluation. GPT-4V is a GPT4 variant that can directly perceive image inputs, showing state-of-the-art image understanding performance.

4.2 Datasets

Seven datasets covering NLP, ASR and VQA are used in this paper. For text-only ICL, four datasets are used in four different task categories: the TREC dataset for topic classification (Voorhees and Tice, 2000), the SST2 dataset for sentiment analysis (Socher et al., 2013), the Spider dataset for text-to-SQL (Yu et al., 2018), and the CHiME-4 (Vincent et al., 2017) split of the HyPoradise dataset (Chen et al., 2023) for generative language model re-scoring to correct pre-generated ASR transcriptions. For audio-text ICL, Two datasets are used for ASR tasks, namely RASC863 (ChineseLDC.org, 2004) and CORAAL (Gunter et al., 2021). RASC863 is a commonly used Chinese dialect ASR dataset and its dialectal words split of Chongqing and Guangzhou dialects are used. CORAAL is an English corpus with speech recordings from regional African Americans. For image-text ICL, VQA experiments are conducted on OKVQA (Marino et al., 2019), a dataset that requires methods to draw upon external knowledge to answer the visual questions.

4.3 Baselines

On all three modalities, **random selection** and improved **KATE** (Liu et al., 2022) are used as baseline approaches. For random selection, in-context examples are uniformly selected from the example datastore three times and the average results are reported. For KATE (Liu et al., 2022), k neighbours that are nearest to the test input in the embedding space in terms of Euclidean distance are selected. For ASR ICL, the encoder of Whisper large-v2 acts as the embedding retrieval module on the Chinese dataset, while on the English dataset, we use the encoder of Whisper large-v3. In text-ICL, OpenAI

Setting	Corpus & In-context example number k								
	RASC863 Chongqing				RASC863 Guangzhou				CORAAL <15s
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$
random	67.1	56.1	52.7	51.0	61.7	38.3	31.2	28.8	13.2
KATE+	67.1	54.7	51.3	49.7	61.3	36.1	**26.9**	**24.8**	12.6
ByCS	**62.4**	**53.4**	**50.6**	**48.6**	**49.5**	**31.9**	27.1	26.6	**12.4**
oracle ByCS	62.4	52.4	49.5	47.2	49.4	30.7	25.8	24.7	12.4

(a) Results with Whisper-large-v2

Setting	Corpus & In-context example number k								
	RASC863 Chongqing				RASC863 Guangzhou				CORAAL <15s
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$
random	68.9	60.3	57.0	55.7	67.1	42.8	38.3	35.2	12.4
KATE+	68.1	58.2	54.8	54.1	67.7	41.3	34.3	31.6	12.1
ByCS	**63.5**	**56.3**	**53.5**	**51.8**	**50.7**	**36.7**	**33.0**	**31.5**	**12.0**
oracle ByCS	63.4	55.2	53.0	50.7	51.3	35.6	31.9	30.7	11.9

(b) Results with Whisper-large-v3

Table 1: %WERs on RASC863 dialectal word dataset and CORAAL with different in-context example selection methods. For RASC863, the example datastore is the RASC863 dialectal word dataset of the corresponding dialect. For CORAAL, the size of the example datastore for ByCS is narrowed down to 10 using kNN algorithm. For the "oracle ByCS" setting, the ground-truth label $\mathbf{Y}_{\text{ref}}$ is used in the inverse reference.

`text-embedding-ada-002` is used as the embedding retrieval model. For VQA ICL, KATE is only based on the embedding space of the query image and EVA02-CLIP-bigE-14-plus (Sun et al., 2023b) serves as the embedding retrieval module. We use the term "KATE+" to refer to the baseline in our paper, putting stress on the fact that it is actually an improved KATE version enhanced using stronger embedding retrieval models, which results in better performance. For text ICL, **bm25** (Robertson et al., 1995) and **LLM-R** (Wang et al., 2023b) are also compared as baselines. bm25 is a ranking metric originally designed for search engines to estimate the relevance of documents to a given query based on word-overlapping similarity. LLM-R provides a recent and preferment dense retriever distilled using a reward model trained based on LLM feedback.

5 Results

5.1 ASR ICL

Results in WER are reported for ASR tasks in Table 1, and here in Chinese WER is calculated based on Chinese characters, which is also termed as character error rate.

The ByCS method outperforms the KATE+ baseline in most cases, showing the robustness and effectiveness of our method. When the number of in-context examples k is small, ByCS surpasses KATE+ baseline in a large margin, with a 10.25% relative WER reduction on average when $k = 1$. Such performance advantage of ByCS reduces when the number of in-context examples increases, which may be attributed to the fact that ByCS performs the inverse inference of each in-context example individually by applying an independence assumption that ignores the contextual interactions between different in-context examples. The use of $\mathbf{Y}_{\text{ref}}$ in "oracle ByCS" further boosts the performance gain, indicating the upper bound of our method with the same number of k.

5.2 Ablation study on ASR ICL

5.2.1 Inverse decoding option

The influence of different decoding options of inverse inference is studied on the RASC863 dialectal word dataset. The results are shown in Table 2. For the setting notation, "noprompt" denotes decoding in the default decoding option, and "prompt" means to decode with a specially designed *prompt* "识别方言" (meaning to "recognize dialect speech"). "LID" denotes decoding with the correct language identity of Chinese ("zh").

The results show that among the three inverse de-

coding options, "noprompt" obtains the best performance, "prompt" becomes the second, and "LID" the worst. The WERs of inverse inference are reported in Table 3. The WERs under the "noprompt" setting are more than 100% due to the high insertion error rate. The repeated outputs are not removed when calculating the WERs of inverse inference and when calculating the text similarity, making a more obvious distinction between the examples with high mutual information interaction and those with low.

Although it may be a little counter-intuitive that low inverse inference accuracy results in high ByCS selection performance, it is reasonable since inverse inference in ByCS helps to separate good in-context examples from the rest, which can be better achieved by using worse decoding options during inverse inference. This is because our decoding options can often make the model make more mistakes for worse in-context examples.

Setting		Corpus	
Text similarity measurement	Inverse decoding option	RASC863 Chongqing	RASC863 Guangzhou
Jaccard coefficient	noprompt	**62.4**	**49.5**
	prompt	62.9	50.7
	LID	64.1	52.3
BERT wordvecs	noprompt	**62.4**	51.5
	prompt	63.5	56.8
	LID	64.5	57.7

Table 2: %WERs of Whisper large-v2 on RASC863 dialectal word dataset using ByCS method with different inverse decoding options and text similarity measurements. The number of in-context examples is $k = 1$.

Inverse decoding option	Corpus	
	RASC863 Chongqing	RASC863 Guangzhou
noprompt	91.5	125.2
prompt	70.2	70.1
LID	54.6	61.7

Table 3: Inverse inference %WERs of Whisper large-v2 on RASC863 dialectal word dataset with different inverse decoding options.

Setting	In-context example number k			
	$k=1$	$k=2$	$k=3$	$k=4$
KATE+	67.1	54.7	51.3	49.7
$\text{ByCS}_{\text{largev2}}$	**62.4**	53.4	50.6	**48.6**
$\text{ByCS}_{\text{small}}$	64.2	**53.3**	**50.5**	48.7

(a) Results with Whisper large-v2

Setting	In-context example number k			
	$k=1$	$k=2$	$k=3$	$k=4$
KATE+	68.1	58.2	54.8	54.1
$\text{ByCS}_{\text{largev3}}$	**63.5**	**56.3**	**53.5**	51.8
$\text{ByCS}_{\text{small}}$	64.4	56.5	54.1	**51.7**

(b) Results with Whisper large-v3

Table 4: %WERs on RASC863 Chongqing dialectal word dataset with ByCS with different inverse inference models. $\text{ByCS}_{\text{largev3}}$ and $\text{ByCS}_{\text{small}}$ use Whisper-large-v3 and Whisper-small as the inverse inference model separately.

5.2.2 Text similarity measurement

The results of ByCS with different text similarity measurements are also reported in Table 2. For the setting notation, the "Jaccard coefficient" is a commonly used statistic to gauge similarity, defined as the intersection over the union of two sentences. "BERT wordvecs" is to measure similarity based on the Euclidean distance in the embedding space of BERT encoded word vectors. The embedding retrieval module is `bert-base-chinese` [1].

ByCS with the Jaccard coefficient as text similarity have lower WERs, which may be because the training data of the BERT model doesn't include sufficient dialectal Chinese words and expressions. It also indicates that ByCS can work well with even a simple rule-based text similarity measurement, further verifying its high robustness. The Jaccard coefficient is used as the text similarity measurement in later experiments unless explicitly specified, due to the performance and simplicity.

5.2.3 Inverse inference model

The inverse inference with different models is also investigated, with the results displayed in Table 4. A smaller model is used for inverse inference to speed up ByCS, since it is expensive to perform inverse inference using the inference model for every candidate example in datastore. Replac-

[1] https://huggingface.co/bert-base-chinese

Setting	Corpus & In-context example number k								
	TREC(%Acc. ↑)			SST2(%Acc. ↑)		Spider(%Acc. ↑)	HyPoradise CHiME-4 (%WER ↓)		
	$k=1$	$k=2$	$k=4$	$k=1$	$k=2$	$k=1$	$k=1$	$k=2$	$k=5$
default		63.0			92.92	67.41		8.0	
random	63.5	72.7	75.3	94.96	94.80	67.02	7.5	7.5	7.3
KATE+	78.8	86.4	**91.0**	95.05	94.69	69.44	7.7	7.1	6.8
bm25	74.6	**89.4**	89.8	**95.27**	**95.40**	67.41	7.4	7.5	8.1
LLM-R	78.0	88.8	90.4	95.05	94.02	67.82	7.4	6.9	7.0
ByCS	**81.2**	88.0	90.6	95.16	95.04	**69.63**	**7.1**	**6.8**	**6.4**

(a) Results using GPT-3.5-Turbo

Setting	Corpus & In-context example number k								
	TREC(%Acc. ↑)			SST2(%Acc. ↑)		Spider(%Acc. ↑)	HyPoradise CHiME-4 (%WER ↓)		
	$k=1$	$k=2$	$k=4$	$k=1$	$k=2$	$k=1$	$k=1$	$k=2$	$k=5$
default		75.2			95.01	69.63		11.6	
random	81.3	82.5	84.6	96.38	96.11	70.66	6.9	6.8	6.5
KATE+	88.2	91.6	93.4	96.43	95.85	71.95	7.0	**6.3**	**5.8**
bm25	81.8	87.4	91.4	96.19	96.09	71.47	6.8	6.6	6.3
LLM-R	88.2	91.0	**93.6**	95.74	95.06	72.63	6.8	**6.3**	5.9
ByCS	**88.6**	**92.4**	**93.6**	**96.55**	**96.31**	**72.82**	**6.7**	**6.3**	5.9

(b) Results using GPT-4

Table 5: Results of four text ICL tasks on two GPT-family models with different in-context example selection methods. The evaluation metrics are denoted in the brackets. The example datastore is narrowed down to a small size using kNN for ByCS. In the 'default' setting, the answers are generated directly with the questions without ICL.

ing Whisper-large-v2/v3 with Whisper-small will speed up six times[2]. For the notation, the subscript denotes the inverse inference model. For example, $\text{ByCS}_{\text{small}}$ is the ByCS method with Whisper small as an inverse inference model.

$\text{ByCS}_{\text{small}}$ has similar results to $\text{ByCS}_{\text{largev2}}$ and $\text{ByCS}_{\text{largev3}}$, verifying the effectiveness of using a smaller model from the same family for inverse inference. This is intuitive since Whisper-small is trained using the same data and settings compared to the inference model Whisper-large-v2 and Whisper-large-v3, which therefore processes information similarly and can serve as a good alternative when evaluating the quality of the in-context examples. The smaller size of Whisper-small makes ByCS a more practical method in cost-sensitive scenarios. A detailed analysis of time cost is in Appendix B.

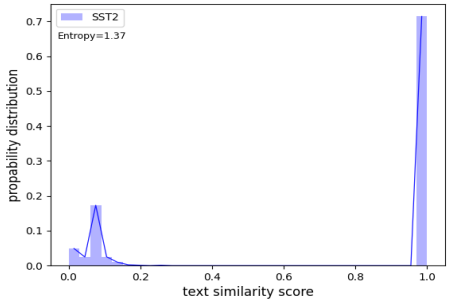
5.3 Text ICL

Text-only ICL results are shown in Table 5. As shown, ByCS outperforms all baselines on most dataset settings, showing not only the effectiveness but also the robustness of ByCS. In particular, ByCS outperforms the best baseline on the generative ASR rescoring dataset HyPoradise with a considerable 4.7% relative WER reduction with GPT-3.5-Turbo. On TREC and SST2 datasets, ByCS does not always outperform the baselines. This indicates that ByCS is more suitable for open-ended long-answer datasets due to the calculation of text similarity in ByCS, in which answers are much more diverse and examples with rich information interactions can be better separated. In contrast, in multi-choice classification datasets, only a few short answers are often available, containing little contextual information. As the example shown in Figure 4, the distribution of the text similarity for ranking the examples is often sharp, merging the optimal and the suboptimal examples. Furthermore, considering the hypothesized labels of the test inputs for inverse inference, the hypothesized answers in open-ended datasets (in the form of long sentences) are often more similar to their corresponding references compared to those in the multi-choice classification datasets (in the form of a word or phrase or just an index of choice).

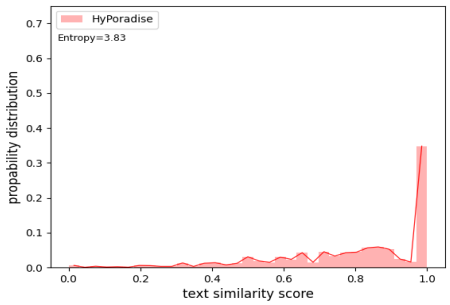
[2] https://github.com/openai/whisper

It is observed that different in-context example selection methods perform differently with different models, even though on the same dataset. The bm25 method outperforms the KATE+ method with GPT-3.5-Turbo on the SST2 dataset, but not with GPT4. Compared to KATE+ and bm25 that is model-free in the actual selection step, the performance advantage of ByCS is more consistent since it takes into account the influence of the model. The outputs of the inverse inference model are used, which can serve as a good approximation to the inference model as verified in Section 5.2.3.

Note that for ByCS on GPT-4, although the inverse inference procedure is conducted on GPT-3.5-Turbo, the performances of ByCS are still superior. This further verifies that smaller models from the same model family can serve as a good low-cost approximation of the inverse inference model.



(a) Distribution on SST2



(b) Distribution on HyPoradise

Figure 4: The distribution of text similarity scores on different datasets. The text similarity score is the Jaccard coefficient. The entropy of distribution is calculated and placed on the upper left. The distribution on the multichoice classification dataset SST2 (blue) is much sharper than that of the open-ended dataset HyPoradise (red).

5.4 VQA ICL

ByCS is tested on VQA ICL and the results are reported in Table 6. ByCS outperforms the KATE+ baseline on the VQA ICL task, demonstrating strong performances across modalities. The performance improvement from ByCS is not as obvious as in audio and text tasks, since the answers of VQA are usually short (usually a word or phrase), lacking sufficient contextual information. ByCS on the VQA dataset suffers from the problem of having sharp text similarity score distributions, similar to the multichoice classification dataset. For ByCS with GPT-4V, inverse inference results on Emu-2 are used to pre-select the candidate examples, and ByCS still outperforms the KATE+ baseline. The performance may be further improved if GPT-4V is also used for inverse inference. This demonstrates that ICL may perform similarly cross models not only on speech and text, but also on images.

In-context example number k	Example selection method	
	KATE+	ByCS
k = 2	**40.47**	40.12
k = 4	45.11	**45.14**

(a) Results with Emu-2

In-context example number k	Example selection method	
	KATE+	ByCS
k = 2	52.54	**52.86**
k = 4	54.00	**54.39**

(b) Results with GPT-4V

Table 6: Results of VQA ICL with different in-context example selection methods and numbers of examples on OKVQA dataset.

6 Conclusion

This paper proposes ByCS, a novel in-context example selection method based on Bayes' theorem, which assumes that contextual information interaction is mutual between the test input and in-context examples and selects high-quality examples based on the inverse inference results. Experiments are performed across three modalities: speech, text, and images, using six different tasks and seven datasets. Results demonstrated the robustness and effectiveness of ByCS. It is also validated that the inverse inference results can be approximated using a smaller model from the same model family, which considerably reduces the computational cost. Moreover, relying on text similarity to rank in-context examples, ByCS is more suitable for open-ended

long-answer datasets which contain sufficient contextual information. Future work is to extend the inverse inference to sequences with multiple in-context examples to model the interactions among the in-context examples.

Limitations

There are three limitations to this work. First, ByCS follows the simple assumption that the influence of each in-context example is independent and treats each in-context example individually, which neglects the contextual interactions between in-context examples. The approximation may not be adapted to the scenario in which the number of in-context examples is high. Second, ByCS requires sufficient contextual diversity to select optimal examples, which depends on text similarity to evaluate inverse inference results. ByCS may suffer a performance penalty when applied to a short-answer dataset. The third limitation is the extra time cost introduced by inverse inference, making ByCS less suitable for cost-sensitive scenarios. Future work includes enhancing ByCS in more scenarios.

Ethics Statement

The work doesn't give rise to any ethical risks and issues. All the models and data used in this paper are publicly accessible and used under licenses.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. In *Proc. NeurIPS*.

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. 2023. Skill-based few-shot selection for in-context learning. *arXiv preprint arXiv:2305.14210*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proc. NeurIPS*.

Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Ensiong Chng. 2023. Hyporadise: An open baseline for generative speech recognition with large language models. In *Proc. NeurIPS*.

ChineseLDC.org. 2004. Introduction to RASC863. *http://www.chineseldc.org/doc/CLDC-SPC-2004-005/intro.htm*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. In *Proc. ACL 2023 findings*.

Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. 2022. Magma–multimodal augmentation of generative models through adapter-based finetuning. In *Proc. EMNLP 2022 findings*.

Kaylynn Gunter, Charlotte Vaughn, and Tyler Kendall. 2021. Contextualizing/s/retraction: Sibilant variation and change in washington dc african american language. *Language Variation and Change*, 33(3):331–357.

Ming-Hao Hsu, Kai-Wei Chang, Shang-Wen Li, and Hung-yi Lee. 2023. An exploration of in-context learning for speech language model. *arXiv preprint arXiv:2310.12477*.

Dan Iter, Reid Pryzant, Ruochen Xu, Shuohang Wang, Yang Liu, Yichong Xu, and Chenguang Zhu. 2023. In-context demonstration selection with cross entropy difference. *arXiv preprint arXiv:2305.14726*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. ICML*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proc. DeeLIO*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022a. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *Proc. ICLR*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022b. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proc. ACL (Volume 1: Long Papers)*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proc. CVPR*.

Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proc. EMNLP*.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

OpenAI. 2023a. Gpt-4 technical report.

OpenAI. 2023b. Gpt-4v(ision) system card.

Jing Pan, Jian Wu, Yashesh Gaur, Sunit Sivasankaran, Zhuo Chen, Shujie Liu, and Jinyu Li. 2023. Cosmic: Data efficient instruction-tuning for speech in-context learning. *arXiv preprint arXiv:2311.02248*.

Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proc. NAACL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP*, pages 1631–1642.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. In *Proc. ICLR*.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, et al. 2023a. Generative multimodal models are in-context learners. *arXiv preprint arXiv:2312.13286*.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023b. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arxiv preprint arXiv:2310.13289*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arxiv preprint arxiv:2307.09288*.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Proc. NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NeurIPS*.

E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer. 2017. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech and Language*, 46:535–557.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proc. SIGIR*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023a. Neural codec language models are

zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Siyin Wang, Chao-Han Huck Yang, Ji Wu, and Chao Zhang. 2023c. Can whisper perform speech-based in-context learning. *arxiv preprint arXiv:2309.07081*.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023d. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Prov. NeurIPS*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. NeurIPS*.

Shangqing Xu and Chao Zhang. 2024. Misconfidence-based demonstration selection for llm in-context learning. *arXiv preprint arXiv:2401.06301*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proc. EMNLP*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proc. EMNLP*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proc. ICML*.

A Experimental Details

A.1 Datasets, baselines and prompt templates

The dataset details are listed in Table 9. For Spider, the evaluation metric is execution accuracy. For CORAAL, we use the processing script from the FairSpeech project[3]. For convenience, we only use speech less than 15 seconds because Whisper can accept input audio up to 30 seconds. For the ASR dataset, there is no train/test split, the dataset except the test input serves as the in-context example datastore. For bm25 implementation, we use the okapi variant in rank_bm25[4] library. The inverse inference example is presented in Figure 5 and prompt templates are shown in Table 13.

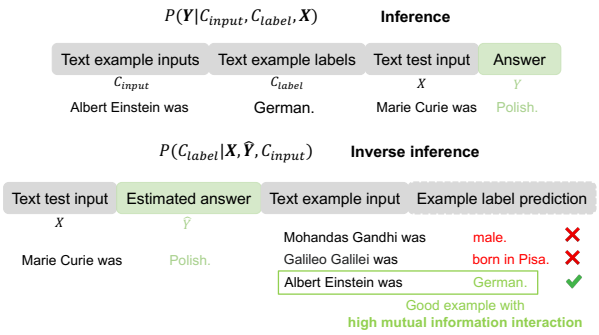


Figure 5: We provide an additional "inverse inference" illustration of the proposed Bayesian example selection method for in-context learning in a text format, similar to Min et al. (2022).

Figure 6: An illustration of the calculation of text similarity between inverse inference results and their true labels in Mandarin accent recognition, where the red inverse inference tokens indicate misrecognition.

[3] https://github.com/stanford-policylab/asr-disparities

[4] https://github.com/dorianbrown/rank_bm25

A.2 First-round inference of ByCS

We experimented with ByCS on different first-round inference settings to examine the influence of first-round inference, and the results are reported in Table 7. The first-round inference produces the hypothesized label of test input. With better first-round inference hypotheses, the approximated inverse inference probability will be more close to the oracle one. Figure 6 provides an example of text similarity calculation. The first-round accuracy for the 'default', 'random' and 'KATE+' settings is 63.0, 75.8 and 91.0, respectively. The first-round inference with ICL improves the accuracy of the hypothesized label, thus boosting the performance of ByCS. In practice, we use ICL with random example selection as the first-round inference setting for ASR ICL and best ICL baseline as the first-round inference setting for text and VQA ICL.

First-round inference	In-context example number k		
	$k=1$	$k=2$	$k=4$
default	75.6	83.8	88.4
random $k=4$	79.8	87.0	91.6
KATE+ $k=4$	81.2	88.0	90.6

(a) Results with GPT-3.5-Turbo

First-round inference	In-context example number k		
	$k=1$	$k=2$	$k=4$
default	87.2	91.8	93.0
random $k=4$	86.6	92.4	93.0
KATE+ $k=4$	88.6	92.4	93.6

(b) Results with GPT-4

Table 7: Results on TREC of ByCS with different first-round inference settings.

A.3 Pre-selection of ByCS

Since the datastore size is usually large, we use a simple ranking algorithm to compress in-context example datastore and then use ByCS inverse inference to select good examples. We usually choose kNN as the ranking algorithm and twice the maximum number of in-context examples as reduced size after pre-selection. For RASC863, we simply use the speech from the same speaker as in-context examples, so the number of reduced size is approximate. We experimented on the TREC dataset to analyze whether reduced size matters, the results are reported in 8. The results imply that reduced

size has nearly negligible impact on the performance of ByCS method. Thus twice the number of in-context examples is a balanced choice for example diversity and conducting speed. The details of pre-selection are shown in Table 9.

Reduced size	In-context example number k		
	$k=1$	$k=2$	$k=4$
4	81.6	87.6	90.6
8	81.2	88.0	90.6
16	81.0	88.0	90.4

(a) results on GPT-3.5-Turbo

Reduced size	In-context example number k		
	$k=1$	$k=2$	$k=4$
4	88.0	92.6	93.2
8	88.6	92.4	93.6
16	88.4	92.8	93.2

(b) results on GPT-4

Table 8: Results on TREC of ByCS with different reduced sizes after pre-selection.

B Analysis of time cost

B.1 Computational complexity

Although ByCS may be time-consuming, the existing improvement methods have reduced the complexity from $\mathcal{O}(N)$ to $\mathcal{O}(1)$, where N is the size of the example datastore. The original version of ByCS will conduct inverse inference on every candidate in the whole dataset, which results in complexity in $\mathcal{O}(N)$. Using a smaller model for fast inverse inference decreases the number of computations by a constant factor. For instance, Whisper small is 6 times faster than Whisper large, and using Whisper small for inverse inference reduces the inverse inference cost by $\sim$6 times. Furthermore, by using a ranking-based pre-selection, we can reduce the size of the example datastore to a fixed number, reducing the computational complexity of inverse inference further down to $\mathcal{O}(1)$. In our experiments, we found empirically that a number around 10 is a good choice in balancing the example diversity and conduction speed, as shown in Appendix A.3.

B.2 Attempt to further speed up

Since inverse inference spends most of its time in ByCS, we try to conduct inverse inference on examples in the datastore before the test input arrives. For each example in the datastore, suitable in-context examples are selected for it using ByCS. In practice, the in-context examples of the test input are those of the nearest neighbour. By this means, the time cost of ByCS is comparable with kNN-based methods. The results of this new sped-up version of ByCS, which is denoted as $\text{ByCS}_{\text{fast}}$ are shown in Table 14. As expected, $\text{ByCS}_{\text{fast}}$ always performs worse than ByCS. Furthermore, $\text{ByCS}_{\text{fast}}$ is more dependent on the contextual diversity. On the open-ended long-answer speech datasets, $\text{ByCS}_{\text{fast}}$ can outperform the best baseline. While on short-answer text datasets, the performance of $\text{ByCS}_{\text{fast}}$ suffers a significant deterioration. It emphasizes the importance of inverse inference directly on test input, not on a similar substitution.

Modality	Task category	Dataset	Train size	Test size	Pre-selection	Reduced size
Text	Topic classification	TREC	5452	500	kNN	8
	Sentiment analysis	SST2	67349	872	kNN	4
	Text to SQL	Spider	7000	1034	kNN	3
	ASR LM rescoring	HyPoradise CHiME-4	9728	1320	kNN	10
Audio	Automatic speech recognition	RASC863 Guangzhou	1889	1990(1.41h)	same speaker	~ 10
		RASC863 Chongqing	2993	2994(3.26h)	same speaker	~ 15
		CORAAL <15s	2761	2762(6.77h)	kNN	10
Image	Vision question answering	OKVQA	9009	5046	kNN	8

Table 9: Datasets used in this work

Dataset	Template example
TREC	Question: What is the temperature at the centre of the earth? Available Type: description, entity, expression, human, number, location. Type: number.
SST2	Review: "The Time Machine" is a movie that has no interest in itself. Available sentiment: positive, negative. Sentiment: negative.
Spider	Given the database schema, you need to translate the question into the SQL query. Database schema: Table name: Movie Creation SQL: CREATE TABLE Movie(mID int primary key, title text, year int, director text) Table name: Reviewer Creation SQL: CREATE TABLE Reviewer(rID int primary key, name text) Table name: Rating Creation SQL: CREATE TABLE Rating(rID int, mID int, stars int, ratingDate date, FOREIGN KEY (mID) references Movie(mID), FOREIGN KEY (rID) references Reviewer(rID)) Question: Find the names of all reviewers who have contributed three or more ratings. SQL query: SELECT T2.name FROM Rating AS T1 JOIN Reviewer AS T2 ON T1.rID = T2.rID GROUP BY T1.rID HAVING COUNT(*) >= 3.
HyPoradise CHiME-4	You need to do language model rescoring in ASR. Given the 5-best hypotheses, you need to report the true transcription from the 5-best hypotheses. The 5-best hypothesis is: interest rates rose on torture and treasury bills sold by the government yesterday at its regular weekly auction. interest rates rose on short-term treasury bills sold by the government yesterday at its regular weekly auction. interest rates rose at a torture and treasury bill sold by the government yesterday at its regular weekly auction. interest rates rose on a torture and treasury bill sold by the government yesterday at its regular weekly auction. interest rates rose on torturing treasury bills sold by the government yesterday at its regular weekly auction. The true transcription from the 5-best hypotheses is: interest rates rose on short-term treasury bills sold by the government yesterday at its regular weekly auction.
OKVQA	 Answer in one word or phrase. What softwood is used to close the top of the container in his hand? cork.

Table 10: Prompt template examples used in this work

In-context example number k	Inverse inference model	Text similarity measurement & inverse decoding option					
		Jaccard coefficient			BERT wordvecs		
		noprompt	prompt	LID	noprompt	prompt	LID
$k=1$	ByCS$_{largev2}$	**62.4**	62.9	64.1	**62.4**	63.5	64.5
	ByCS$_{small}$	64.2	64.0	65.4	65.0	65.4	66.3
$k=2$	ByCS$_{largev2}$	53.4	**53.3**	53.7	53.6	54.1	54.1
	ByCS$_{small}$	**53.3**	53.7	54.0	54.1	54.9	54.8
$k=3$	ByCS$_{largev2}$	50.6	51.0	50.9	**50.2**	51.6	50.6
	ByCS$_{small}$	50.5	50.5	51.1	51.3	50.9	51.3
$k=4$	ByCS$_{largev2}$	**48.6**	48.7	48.7	49.1	48.9	49.1
	ByCS$_{small}$	48.7	48.7	**48.6**	49.6	49.1	49.9

(a) Results with Whisper large-v2

In-context example number k	Inverse inference model	Text similarity measurement & inverse decoding option					
		Jaccard coefficient			BERT wordvecs		
		noprompt	prompt	LID	noprompt	prompt	LID
$k=1$	ByCS$_{largev3}$	**63.5**	64.1	65.6	64.5	65.3	65.8
	ByCS$_{small}$	64.4	64.7	64.8	65.5	65.0	65.6
$k=2$	ByCS$_{largev3}$	**56.3**	**56.3**	57.0	57.7	57.0	57.8
	ByCS$_{small}$	56.5	57.0	57.0	57.3	57.2	57.5
$k=3$	ByCS$_{largev3}$	**53.5**	54.1	53.7	55.2	55.6	54.9
	ByCS$_{small}$	54.1	54.6	54.4	55.5	55.3	55.4
$k=4$	ByCS$_{largev3}$	51.8	52.3	52.1	53.1	53.4	53.3
	ByCS$_{small}$	**51.7**	52.2	51.9	53.6	53.4	53.5

(b) Results with Whisper large-v3

Table 11: Full results on RASC863 Chongqing dialectal word dataset of ByCS with different inverse decoding options, text similarity measurements and inverse inference models. The subscript denotes the inverse inference model.

In-context example number k	Inverse inference model	Text similarity measurement & inverse decoding option					
		Jaccard coefficient			BERT wordvecs		
		noprompt	prompt	LID	noprompt	prompt	LID
$k=1$	$\text{ByCS}_{\text{largev2}}$	**49.5**	50.7	52.3	51.5	56.8	57.7
	$\text{ByCS}_{\text{small}}$	52.9	55.1	58.7	56.8	57.1	58.8
$k=2$	$\text{ByCS}_{\text{largev2}}$	**31.9**	33.6	34.3	32.9	34.3	35.0
	$\text{ByCS}_{\text{small}}$	34.5	34.1	35.6	35.1	35.9	37.0
$k=3$	$\text{ByCS}_{\text{largev2}}$	**27.1**	28.4	27.7	**27.1**	27.4	27.5
	$\text{ByCS}_{\text{small}}$	28.3	27.8	27.6	27.9	28.6	28.3
$k=4$	$\text{ByCS}_{\text{largev2}}$	26.6	25.5	**24.8**	25.4	26.5	25.5
	$\text{ByCS}_{\text{small}}$	25.9	25.7	25.5	25.3	26.3	26.2

(a) Results with Whisper large-v2

In-context example number k	Inverse inference model	Text similarity measurement & inverse decoding option					
		Jaccard coefficient			BERT wordvecs		
		noprompt	prompt	LID	noprompt	prompt	LID
$k=1$	$\text{ByCS}_{\text{largev3}}$	**50.7**	51.8	55.4	56.6	57.1	59.1
	$\text{ByCS}_{\text{small}}$	55.3	55.4	61.7	61.8	58.7	60.7
$k=2$	$\text{ByCS}_{\text{largev3}}$	**36.7**	38.1	38.9	38.2	37.8	38.9
	$\text{ByCS}_{\text{small}}$	37.3	37.3	40.0	39.0	38.0	39.6
$k=3$	$\text{ByCS}_{\text{largev3}}$	**33.0**	33.4	34.0	33.6	33.4	33.3
	$\text{ByCS}_{\text{small}}$	33.3	33.3	34.6	34.8	33.3	34.3
$k=4$	$\text{ByCS}_{\text{largev3}}$	31.5	31.3	31.4	31.7	31.7	31.4
	$\text{ByCS}_{\text{small}}$	**31.0**	31.5	31.9	31.5	**31.0**	**31.0**

(b) Results with Whisper large-v3

Table 12: Full results on RASC863 Guangzhou dialectal word dataset of ByCS with different inverse decoding options, text similarity measurements and inverse inference models. The subscript denotes the inverse inference model.

Test input	KATE+	ByCS
sometime they do not act like they hear nothing but know nothing about tarboro when you say you from tarboro they will talk about where is tarboro at (CORAAL)	Example: in the era and th the way in there them floors along that time they cut timber certain time of the year Result: sometimes it do not work out there but no nothing about tarver when you say you from tarver they will talk about where tarver is	Example: so they put her and him together and i was praying to the lord that he did not try to jump out of there cause i was so scared me and my husband Result: sometimes they do not want to let their hear nothing but know nothing about tarver when you say you from tarver they will talk about where tarver is
What person 's head is on a dime? human. (TREC)	Example: What is money made of? entity. Result: entity.	Example: Who is the head of the World Bank? human. Result: human.

Table 13: In-context examples selected by kNN and ByCS and corresponding results.

Setting	Corpus & In-context example number k								
	RASC863 Chongqing				RASC863 Guangzhou				CORAAL <15s
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$
best baseline	67.1	54.7	51.3	49.7	61.3	36.1	**26.9**	**24.8**	12.6
ByCS$_{\text{fast}}$	63.1	**52.5**	**50.2**	**48.3**	55.8	35.6	29.2	27.1	12.5
ByCS	**62.4**	53.4	50.6	48.6	**49.5**	**31.9**	27.1	26.6	**12.4**

(a) Results with Whisper-large-v2

Setting	Corpus & In-context example number k								
	RASC863 Chongqing				RASC863 Guangzhou				CORAAL <15s
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$
best baseline	68.1	58.2	54.8	54.1	67.1	41.3	34.3	31.6	12.1
ByCS$_{\text{fast}}$	66.7	57.5	54.5	52.6	60.5	40.3	34.1	32.3	12.2
ByCS	**63.5**	**56.3**	**53.5**	**51.8**	**50.7**	**36.7**	**33.0**	**31.5**	**12.0**

(b) Results with Whisper-large-v3

Setting	Corpus & In-context example number k				
	TREC(%Acc. ↑)			SST2(%Acc. ↑)	
	$k=1$	$k=2$	$k=4$	$k=1$	$k=2$
best baseline	78.8	**89.4**	**91.0**	**95.27**	**95.40**
ByCS$_{\text{fast}}$	77.0	83.8	86.4	94.15	94.61
ByCS	**81.2**	88.0	90.6	95.16	95.04

(c) Results using GPT-3.5-Turbo

Setting	Corpus & In-context example number k				
	TREC(%Acc. ↑)			SST2(%Acc. ↑)	
	$k=1$	$k=2$	$k=4$	$k=1$	$k=2$
best baseline	88.2	91.6	**93.6**	96.43	96.11
ByCS$_{\text{fast}}$	85.4	89.2	92.6	95.07	95.18
ByCS	**88.6**	**92.4**	**93.6**	**96.55**	**96.31**

(d) Results using GPT-4

Table 14: Results of ByCS$_{\text{fast}}$ on speech and text tasks. Results of best baseline and ByCS are also shown for comparison.