Z-ICL: ZERO-SHOT IN-CONTEXT LEARNING WITH PSEUDO-DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models (LMs) perform a new task at test time either through zero-shot inference or few-shot in-context learning, i.e., conditioning on the k-shot training data (so-called *demonstrations*). Prior work suggests that in-context learning mainly activates the intrinsic ability of the LM. We argue that this implies zero-shot performance of the LM is underestimated and can be as good as in-context learning if we inform the LM with the correct space of the inputs and the labels using *pseudo-demonstrations*. We also identify an additional factor which we call *the copying effect*: if pseudo-demonstrations includes an input that is very similar to the test input, the model prediction is heavily influenced by the paired label of that input. Putting altogether, we introduce Z-ICL, a new zero-shot prompting method that constructs pseudo-demonstrations without any training data that (a) informs the correct space of the inputs and the outputs and (b) reduces the copying effect so that the prediction is less affected by the pairings in the pseudo-demonstration. Z-ICL includes (a) leveraging nearest neighbors from a raw text corpus and pairing them with random but valid labels and (b) proposing a set of techniques such as *physical neighbors* and *synonym labeling*. Z-ICL outperforms previous zero-shot methods by a significant margin, and is on par with in-context learning with gold training data on a range of text classification datasets. Together, Z-ICL provides a significantly higher estimate of the model's ability to perform a new task zero-shot, and poses a set of new questions about the capacities of LMs.

1 INTRODUCTION

Large language models (LMs) can perform new tasks simply by conditioning on input-label pairs from the training data, so-called *demonstrations* (Brown et al., 2020). This in-context learning (ICL) is significantly better than zero-shot methods that do not use demonstrations. Recent work suggests that in-context-learning demonstrations are primarily specifying the domain and the format that the target task, instead of providing explicit training signal (Reynolds & McDonell, 2021; Xie et al., 2022; Razeghi et al., 2022; Min et al., 2022) (related work discussed in Appendix A).

We argue that this literature implies that current zero-shot performance (with no demonstrations) levels must be significantly underestimated, since all the required information must already be in the model. We introduce **Z-ICL**: **Z**ero-shot **I**n-**C**ontext **L**earning through creating *pseudo-demonstrations*, aiming to achieve results on par with in-context learning from gold demonstrations. *Pseudo-demonstrations* are constructed without any training data to inform the correct space of the inputs and the outputs, which are necessary condition for successful in-context learning (Xie et al., 2020; Min et al., 2022). In particular, based on Min et al. (2022), correct pairings between inputs and outputs in pseudo-demonstrations are not the necessary condition, which makes it possible to construct pseudo-demonstrations in a zero-shot manner. However, we identify a factor called *the copying effect*—our new observation that the LM predictions can be heavily influenced by input-label pairings if the demonstrations include an input that is very close to the test input.

Based on these intuitions, Z-ICL constructs the pseudo-demonstrations that (a) inform the correct input distribution and the label space and (b) avoid *the copying effect*. To satisfy (a), Z-ICL retrieves a set of nearest neighbors from a raw text corpus and assigns a random label to each. To satisfy (b), we propose two techniques—*physical neighbors* and *synonym labeling*—that twist retrieved sentences and pairs labels in a way that reduce copying from the pseudo-demonstrations. Results

Example #1		
Demo 1	I am giving a zero star to symantec for this version.	`great`
Demo 2	**I recommend not to purchase it. This player is not worth any price.**	**`great`**
Demo 3	So far I have no complains with this player.	`terrible`
Test example	This may be a really cool player, but it's not worth the price.	**`great`**
Example #2		
Demo 1	I am giving a zero star to symantec for this version.	`great`
Demo 2	**I recommend not to purchase it. This player is not worth any price.**	**`terrible`**
Demo 3	So far I have no complains with this player.	`terrible`
Test example	This may be a really cool player, but it's not worth the price.	**`terrible`**

Table 1: An illustration of the copying effect hypothesis with *nearest* in-context learning ($k = 3$). The first three lines are demonstrations, and the last line is the test instance. The model prediction in **red**. The model tends to copy the label from the demonstration input that is close to the test input.

on nine classification datasets indicate Z-ICL significantly outperforms a previous zero-shot method (no-demonstrations) consistently across different LMs (Wang & Komatsuzaki, 2021; Black et al., 2022; Brown et al., 2020), and is on par with in-context learning that uses labeled k-shot training data. Together, Z-ICL provides a significantly higher estimate of the ability or current LMs to perform a new task zero-shot, encourages new ways to improve zero-shot performance by designing even better pseudo-demonstrations, and poses a set of new questions about the capacities of LMs.

2 Copying Effect Hypothesis

The demonstrations are typically sampled uniformly at random from the true distribution, e.g., the training data. We observe that, when demonstrations contain input text that is very similar to the test input, the model exhibits a behavior which we call the *copying effect*. To study this, we evaluate **ICL-gold** (standard ICL) and **ICL-random**; both are ICL methods that use k randomly sampled examples from the training data with gold and random labels, respectively. We then evaluate **nearest ICL-gold** and **nearest ICL-random**, which retrieves the k nearest neighbors for each test input from the training data and assign gold labels and random labels, respectively. We use channel GPT-J (Wang & Komatsuzaki, 2021) as the LM and SimCSE (Gao et al., 2021) for choosing the nearest inputs.

Figure 1 shows that ICL-gold and ICL-random achieve relatively comparable performance, which is consistent with Min et al. (2022) that the correctness of labels in the demonstrations matters much less than we thought. However, with nearest ICL, using random labels is significantly worse than using gold labels. This indicates that the correctness of labels matters significantly more when the inputs in the demonstrations are closer to the test input.

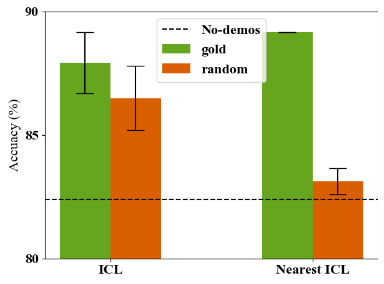


Figure 1: Performance of ICL and nearest ICL, each with gold labels and random labels. Evaluated on three datasets (CR, Amz, Yelp). The gap between gold and random labels is greater with nearest ICL than with ICL, indicating that the correctness of labels matters more when the demonstrations are closer to the test input.

Based on our observation, we define a ***copying effect hypothesis***: the model prediction is heavily biased toward the labels paired with inputs in the demonstrations that are very similar to the test input, which resembles *copying*. Table 1 provides an example. The second input in the demonstrations is very close to the test input both lexically and semantically, and the model prediction tends to follow the label paired with the second input, regardless of what that label is. To better quantify the copying effect, we design an experiment where the demonstrations include an example that is *identical* to the test input, either with a correct label or with an incorrect label. We then see how many times the LM makes a prediction that is the same as the label paired with the identical demonstration example. LM predicts the same label as the one paired with the identical input for over 90% of the times when the label is correct, and over 70% of the times when the label is incorrect, consistently over different LMs (see Appendix C.2). In the next section, we design a zero-shot method where the copying effect can be problematic, and propose new techniques that reduce the copying effect.

3 Our Method: Z-ICL

We introduce **Z-ICL**, a new **Z**ero-shot **In-C**ontext **L**earning method, which predicts the correct label for a given test input x and its candidate classes $\mathcal{Y}$ from a task. Unlike prior methods (Liu et al., 2021; Rubin et al., 2021; Liu et al., 2022) where the target domain and labeled training data of the task are available, Z-ICL constructs pseudo-demonstrations—pairs of inputs and labels—in a zero-shot fashion by leveraging a raw text corpus $\mathcal{C}$, and perform in-context learning. Pseudo-demonstrations are designed (a) to inform the correct input distributions and the correct label space, and (b) to reduce the copying effect (Section 2) so that the model is less affected by incorrectly paired labels.

Step 1: Retrieve Relevant Sentences. Z-ICL retrieves k from $\mathcal{C}$ that are similar to x. Let $s : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ be a similarity function between two sentences, and $\mathcal{N}_k(x)$ be a set of k sentences retrieved from $\mathcal{C}$ with the highest $s(c_i, x)$. While it is possible to construct pseudo-demonstrations directly using $\mathcal{N}_k(x)$, it is highly likely to suffer from the copying effect (Section 2), since retrieved sentences are too similar to the test input. Therefore, we propose a method called **physical neighbor**. Instead of directly using $\mathcal{N}_k(x)$, it selects the sentence that is physically adjacent in $\mathcal{C}$ to each sentence in $\mathcal{N}_k(x)$ as $x_1, x_2 ... x_k$. This method allows $x_1, x_2 ... x_k$ to share similar distribution as x, while being sufficiently distant from x since they are not the k nearest neighbors of x.

Step 2: Construct pseudo-demonstrations. Once $x_1 ... x_k$ are obtained, Z-ICL pairs each x_i with a random label following the intuition from Min et al. (2022). Simply assigning the random label from the candidate set $\mathcal{Y}$ would not achieve the best performance because the LM may find similar sentences from $x_1 ... x_k$ and follow their labels according to the copying effect (Section 2). We therefore propose a technique called **synonym labeling**: we use synonyms of the labels and pair $x_1 ... x_k$ with them, instead of the original labels that will be used for the prediction. Formally, for each x_i, Z-ICL chooses a label $y_i \in \mathcal{Y}$ uniformly at random, and create a pair $(x_i, \tilde{y}_j)$, where $\tilde{y}_j$ is a pre-defined synonym of y_j. This technique (1) sufficiently informs the correct semantic space of the labels, and (2) prevents the copying effect by not having the exact same words as the test labels. Note that the original candidate set $\mathcal{Y}$ is used during the test prediction.

Step 3: Inference. Finally, Z-ICL uses in-context learning by concatenating $(x_1, \tilde{y}_1)$, $(x_2, \tilde{y}_2)$, $\cdots, (x_k, \tilde{y}_k)$ as well as the test input x, feeds it to the LM, and obtains the prediction via $\mathrm{argmax}_{y \in \mathcal{Y}} P(y \mid x_1, \tilde{y}_1, \cdots, x_k, \tilde{y}_k, x)$.

4 Experiment

4.1 Experimental Setup

Text corpus. We use the Demix corpus from Gururangan et al. (2021), a raw text corpus that is not designated for any downstream task. It consists of 16 diverse domains, listed in in Appendix B.1.

Evaluation datasets. We evaluate our methods on nine single-sentence classification datasets: CR (Ding et al., 2008), Amz (Zhang et al., 2015), Amz5 (Zhang et al., 2015), Yelp (Zhang et al., 2015), Yelp5 (Zhang et al., 2015), Tweet-Eval (Barbieri et al., 2020), MR (Pang & Lee, 2004), SST2 (Socher et al., 2013) and SST5 (Socher et al., 2013). Six out of the nine datasets are from domains that are represented in our corpus, while the other three (MR, SST2, and SST5) are not. This split allows us to measure domain coverage effects. See Appendix B.1 for data statistics.

Baselines. We compare Z-ICL with the following methods: (1) No-demonstrations (No-demos), a previously-used zero-shot method, (2) Naive Z-ICL, a version of Z-ICL that uses pseudo-demonstrations but without considering the copying effect, (3) ICL-gold, an in-context learning method from Brown et al. (2020), and (4) ICL-random, a variant of in-context learning that uses random labels from Min et al. (2022). See Appendix B.2 for detailed descriptions of each baseline. Note that (3) and (4) use training data, thus not comparable with Z-ICL.

We use three LMs: GPT-J (Wang & Komatsuzaki, 2021), GPT-NeoX (Black et al., 2022) and GPT-3 (Brown et al., 2020) of sizes 6B, 20B, and 175B, respectively. We use two inference methods: direct (a regular inference used in Brown et al. (2020)) and channel (Min et al., 2021). The similarity function s in Z-ICL is defined as a cosine similarity between SimCSE embeddings (Gao et al., 2021). More implementation details are provided in Appendix B.3.

Model	Channel GPT-J	Direct GPT-J	Channel GPT-NeoX	Channel GPT-NeoX	Channel GPT-3	Direct GPT-3
	$\mathcal{C} \mid !\mathcal{C}$	$\mathcal{C} \mid !\mathcal{C}$	$\mathcal{C} \mid !\mathcal{C}$	$\mathcal{C} \mid !\mathcal{C}$	$\mathcal{C} \mid !\mathcal{C}$	$\mathcal{C} \mid !\mathcal{C}$
Majority	$38.0_{0.0} \mid 40.5_{0.0}$	$38.0_{0.0} \mid 40.5_{0.0}$	$38.0_{0.0} \mid 40.5_{0.0}$	$38.0_{0.0} \mid 40.5_{0.0}$	$47.6_{0.0} \mid 50.0_{0.0}$	$47.6_{0.0} \mid 50.0_{0.0}$
No-demos	$61.0_{0.0} \mid 51.3_{0.0}$	$54.8_{0.0} \mid 43.8_{0.0}$	$43.7_{0.0} \mid 48.1_{0.0}$	$42.8_{0.0} \mid 38.8_{0.0}$	$69.5_{0.0} \mid 80.8_{0.0}$	$72.7_{0.0} \mid 73.2_{0.0}$
Naive Z-ICL	$58.5_{0.6} \mid 56.3_{0.6}$	$59.9_{0.6} \mid 53.9_{0.3}$	$55.1_{0.7} \mid 55.1_{0.7}$	$62.2_{0.7} \mid 59.8_{0.9}$	-	-
Z-ICL (Ours)	$\mathbf{65.8_{0.3}} \mid \mathbf{67.7_{0.3}}$	$\mathbf{61.6_{0.3}} \mid \mathbf{64.8_{0.3}}$	$\mathbf{62.5_{0.6}} \mid \mathbf{60.2_{0.3}}$	$\mathbf{65.4_{0.4}} \mid \mathbf{68.4_{0.6}}$	$\mathbf{73.4_{0.6}} \mid \mathbf{82.4_{74.7}}$	$\mathbf{72.7}_{0.3} \mid \mathbf{78.1_{0.1}}$
Oracles						
ICL-gold	$67.9_{1.7} \mid 72.6_{0.9}$	$65.4_{4.9} \mid 72.7_{4.0}$	$65.9_{1.7} \mid 72.1_{0.9}$	$64.4_{5.2} \mid 73.5_{3.0}$	$73.9_{3.0} \mid 88.1_{1.1}$	$79.3_{2.5} \mid 94.2_{0.2}$
ICL-random	$67.4_{1.5} \mid 71.5_{1.1}$	$63.2_{5.1} \mid 68.6_{5.6}$	$63.7_{1.8} \mid 71.4_{1.2}$	$63.3_{4.9} \mid 65.2_{10.2}$	$72.3_{3.0} \mid 84.8_{1.2}$	$77.4_{2.7} \mid 93.9_{0.5}$

Table 2: Average results with GPT-J and GPT-NeoX. *Oracle* indicates the method has access to the training data, thus is not comparable with the rest of the models. $\mathcal{C}$ and $!\mathcal{C}$ indicate the average results of the datasets covered by $\mathcal{C}$ and not covered by $\mathcal{C}$, respectively. Z-ICL is significantly better than previous zero-shot (No-demos) on all datasets, and is on par with ICL-gold on datasets covered by $\mathcal{C}$.

4.2 Results

Results are reported in Table 2 (results on each dataset in Appendix C.1). First, No-demos outperforms the majority baseline but lags behind ICL-gold or ICL-random that access the training data, confirming the previous work. Constructing the pseudo-demonstrations using the text corpus significantly helps, e.g., Naive Z-ICL is better than No-demos in many cases, but is still worse than ICL-gold. Finally, Z-ICL significantly outperforms all baselines, improving performance by 5–30% absolute over the existing zero-shot method (No-demos), consistently over all datasets and all LMs.

Compared to oracle baselines that access the training data (ICL-gold and ICL-random), Z-ICL performs on par on datasets covered by $\mathcal{C}$, despite being zero-shot. On datasets that are not covered by $\mathcal{C}$, Z-ICL still lags behind ICL-gold and ICL-random. This indicates the importance of the coverage of $\mathcal{C}$; Appendix C.3 shows improving the coverage of $\mathcal{C}$ improves performance on these datasets.

Effect of the copying effect reduction. We quantify the effect of the copying effect reduction by comparing two retrieval methods—(1) **nearest**, a naive retrieval method without physical neighbor, and (2) **physical neighbor**—and comparing three label assignment methods—(1) using the **original** test labels, (2) using **random words**, and (3) using the **synonyms** of the test labels. Results[1] are reported in Figure 2. First, 'physical neighbor' significantly outperforms 'nearest', indicating reducing the copying effect through physical neighbor is critical. Second, using random words is consistently better than using the original labels, indicating that not using words from original test labels is important. Finally, using synonyms is consistently better than using random words, indicating that informing the semantic space of the labels is still important. All in all, these indicate that both physical neighbor and synonym labeling are critical in Z-ICL. See additional experiment that further quantifies the copying effect reduction in the Appendix C.2. More ablations can be found in Appendix C.3.

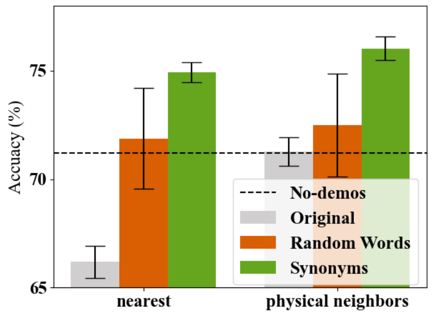


Figure 2: **Effect of copying effect reduction.** The retrieval method *physical neighbor* outperforms *nearest*, and synonym labeling is critical over both retrieval methods.

5 Conclusion

We introduced Z-ICL, a zero-shot in-context learning method that constructs pseudo-demonstrations from a raw text corpus. Z-ICL constructed pseudo-demonstrations to inform the correct space of the inputs and the outputs (following intuition from literature), and to avoid the copying effect (a new behavior identified in this paper). On nine classification datasets, Z-ICL significantly outperforms the previous zero-shot baseline and performs on par with the k-shot demonstrations.

[1]Ablations are on a subset of 6 datasets (CR, Amz5, Yelp5, Tweet, MR, and SST2) with channel GPT-J unless specified otherwise. More ablations are provided in Appendix C.3.

REFERENCES

Anonymous. Overthinking the truth: Understanding how language models process false demonstrations. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. under review.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *EMNLP*, 2020.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022.

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. Language models are few-shot learners. In *NeurIPS*, 2020.

Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, 2008.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. Demix layers: Disentangling domains for modular language modeling. In *NAACL*, 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2021.

Yanchen Liu, Timo Schick, and Hinrich Schütze. Semantic-oriented unlabeled priming for large-scale language models. *arXiv preprint arXiv:2202.06133*, 2022.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. In *ACL*, 2021.

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

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, 2004.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022.

Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *NAACL*, 2021.

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

Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *ICLR*, 2022.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.

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

A Related Work

Demonstrations in ICL. A series of prior work suggests that ICL primarily exposes model functionality that was learned during pre-training. Reynolds & McDonell (2021) suggests that ICL mainly functions by activating the LM's ability obtained during pretraining, and that the LM can achieve significantly better zero-shot performance by using a better template. Xie et al. (2022) shows that ICL can be explained as Bayesian inference for which demonstrations provide noisy evidence. In closed-set tasks, Min et al. (2022) shows that ICL benefits mainly from the correct distribution of the inputs and the labels rather than the input-label correspondence.

Our work draws intuitions from these studies and introduces a better zero-shot method by forming pseudo-demonstrations that are proxies of the input distribution and the label space and better expose the intrinsic ability of the LM.

Better Demonstrations through Retrieval. Prior work has found that, in the setting where large training data is available, choosing demonstration examples that are close to the test input significantly helps in-context learning. Liu et al. (2021) retrieves the nearest training examples to the test input using a sentence encoder, either unsupervised or supervised. Rubin et al. (2021) trains a retrieval system to choose examples that improve in-context learning. Liu et al. (2022) retrieves the nearest neighbors from unlabeled training data, assigns silver labels, and uses them for in-context learning. We similarly use nearest neighbor search to retrieve sentences close to the test input, but are the first to (1) retrieve from a text corpus, in contrast to prior work that uses labeled or unlabeled training data collected for the task, and (2) more closely study the connection between nearest neighbor inputs and random labels, through our copying effect hypothesis.

B Setup Details

B.1 Data Statistics

Corpus. We take the same English corpus from Gururangan et al. (2021) covering 16 diverse domains: 1B, CS, LEGAL, MED, WEBTEXT, REALNEWS, REDDIT, REVIEWS, ACL PAPERS, BREAKING NEWS, CONTRACTS, CORD-19, GITHUB, GUTENBERG, TWEETS, and YELP REVIEWS. See the descriptions and statics in Table 3. For each domain, we 1) subsample 10M paragraphs if the data is larger, 2) split each paragraph into sentences, and 3) remove duplicate sentences while keeping the ordering of the sentences as in the original paragraphs.

Evaluation datasets. Statistics and descriptions of our evaluation datasets are reported in Table 4. For each dataset, we subsample 2000 test examples uniformly at random if the test data is larger, due to limited computational resources.

Domain	Description	#sentences
1B	NewsWire sentences	1.0M
CS	full-text CS papers from S2ORC	1.0M
LEGAL	U.S. court opinions, 1658 to 2018	3.0M
MED	full-text medical papers from S2ORC	1.0M
WEBTEXT	Web documents	2.1M
REALNEWS	articles from REALNEWS	1.8M
REDDIT	Reddit comments from pushshift.io	2.6M
REVIEWS	Amazon product reviews	3.1M
ACL PAPERS	NLP papers from ACL	46K
BREAKING NEWS	latest articles from 400 English news sites	0.5M
CONTRACTS	commercial legal contracts	47K
CORD-19	excerpts from COVID-19 research papers	0.9M
GITHUB	public Github repository contents	0.6M
GUTENBERG	copyright-expired books	0.9M
TWEETS	English tweets from 2013-2018	0.8M
YELP REVIEWS	Yelp restaurant reviews	7.5M

Table 3: List of domains from Gururangan et al. (2021).

B.2 BASELINES

No-demonstrations (No-demos) predicts $\text{argmax}_{y \in \mathcal{Y}} P(y \mid x)$ without using any demonstrations. This is a previously-used zero-shot method (Radford et al., 2019; Brown et al., 2020).

Random inputs selects $x_1...x_k$ from $\mathcal{C}$ uniformly at random, without considering the similarity score with x. It then pairs each x_i with a random label from $\mathcal{Y}$ and uses in-context learning as in Section 3. This baseline uses pseudo-demonstrations, but does not consider the similarity between the test input and the pseudo-demonstrations.

Naive Z-ICL is a version of Z-ICL that uses the most naive retrieval method without the physical neighbor adjustment (Section 3) or synonym labeling (Section 3). This method encourages the *relevance* of the pseudo-demonstrations the most, but does not reduce the copying effect.

We also compare with methods that use the training data, and call them *Oracle* baselines.

ICL-gold (Oracle) uses k input-label pairs from the training data and in-context learning. This is equivalent to the standard in-context learning, first proposed by Brown et al. (2020).

ICL-random (Oracle) uses k inputs from the training data and pairs each input with a random label sampled from $\mathcal{Y}$ uniformly at random, and uses in-context learning (Min et al., 2022).

B.3 EXPERIMENTAL DETAILS

All implementations are done in PyTorch (Paszke et al., 2019). We use int8 quantization (Zeng et al., 2022) to run GPT-NeoX on 40GB A100 machines.

Language models. We experiment with three casual language models: GPT-J (Wang & Komatsuzaki, 2021), GPT-NeoX (Black et al., 2022) and GPT-3 (Brown et al., 2020) of sizes 6B, 20B, and 175B, respectively. We use two inference methods: direct (a regular inference used in Brown et al. (2020)) and channel (Min et al., 2021).

Similarity function. We define a similarity function s to be a cosine similarity between two sentence embeddings, obtained through SimCSE (Gao et al., 2021).[2]

Implementation details. For GPT-J and GPT-NeoX, we use 5 random seeds and report an average and standard deviation. For GPT-3, we use 2 random seeds and only evaluate on five datasets (CR, Amz, Yelp, Tweet, and SST2) due to limited access. If the dataset includes more than 2,000 test examples, we subsample 2,000 examples uniformly at random due to limited computing resources,

[2]In our initial experiments, we explored multiple embedding methods and found SimCSE works the best.

Dataset	# examples	labels	synonyms
Datasets covered by $\mathcal{C}$			
CR	2,000	"terrible", "great"	"bad", "good"
Amz	1,000	"negative", "positive"	"bad", "good"
Amz5	100,050 $\rightarrow$ 2,000	"terrible", "bad", "okay", "good", "great"	"horrible", "negative", "neutral", "positive", "excellent"
Yelp	7,600 $\rightarrow$ 2,000	"negative", "positive"	"bad", "good"
Yelp5	50,000 $\rightarrow$ 2,000	"terrible", "bad", "okay", "good", "great"	"horrible", "negative", "neutral", "positive", "excellent"
Tweet	2,000	"negative", "neutral", "positive"	"bad", "normal", "good"
Datasets not covered by $\mathcal{C}$			
MR	2,000	"terrible", "great"	"bad", "good"
SST2	872	"terrible", "great"	"bad", "good"
SST5	2,210 $\rightarrow$ 2,000	"terrible", "bad", "okay", "good", "great"	"horrible", "negative", "neutral", "positive", "excellent"

Table 4: Statistics of evaluation datasets as well as their labels and synonyms.

Method	Demo	Corpus	Similar	No-Copy
No-demos	-			
Random inputs	pseudo	✓		
Naive Z-ICL	pseudo	✓	✓	
Z-ICL (Ours)	pseudo	✓	✓	✓
ICL-gold (Oracle)	k-shot			
ICL-random (Oracle)	k-shot			

Table 5: Comparison between Z-ICL and baselines. '*Demo*' indicates the type of the demonstrations, either the k-shot training data (k-shot) or constructed from a raw corpus only (pseudo). '*Corpus*' indicates whether an external corpus is used. '*Similar*' indicates whether a similarity function is used. '*No-Copy*' indicates whether the method is designed to reduce the copying effect.

following prior work (Zhao et al., 2021). We use $k = 16$ for all experiments. We use minimal templates from Zhao et al. (2021) without template engineering, e.g., prepending `Review:` and `Sentiment:` to the input and the label, respectively, on a review sentiment classification dataset.[3]

Format of the demonstrations. We use $k = 16$ demonstration examples for all the baselines and methods, unless specified otherwise. We truncate each demonstration example to have up to 256 tokens and the concatenation of them to have up to 1,024 tokens.

Nearest neighbor search. We use SimCSE (Gao et al., 2021) to embed the corpus and the test inputs. We use FAISS (Johnson et al., 2019) to build an index for the corpus offline and perform nearest neighbor search at inference.

Synonym labeling. We manually choose a synonym of each label to perform synonym labeling. A full list of synonyms is reported in Table 4.

C Additional Results

C.1 Main Experiment

Figure 6 and 7 provides the full version of Table 2 with per-dataset numbers for 9 datasets evaluated on 6 LMs.

C.2 Copying Effect Reduction

Copying Label of Identical Example. For the experiment where we put an identical example of the test input as one of the demonstrations described in Section 2, Figure 8 shows LMs tends to copy the label of the identical demonstration.

Quantifying the Copying Effect. In order to ensure that the gains are from avoiding the copying effect, we follow Anonymous (2023) in (1) identifying some induction heads in the Transformer layers that are most responsible for copying, and (2) zero-ing their weights out. If this leads to performance improvements, it is a strong indicator that the method has been suffering from the copying effect. Figure 4b reports results. First, all methods have performance improvements by zero-ing out the induction heads, indicating that all of them suffer from the copying effect to a certain degree. We then find that (1) physical neighbor is affected much less than nearest, and (2) methods with synonym labeling are affected much less than their counterpart without synonym labeling. These are aligned with our intuition that using physical neighbor and synonym labeling help reducing the copying effect.

[3]We did not use templates for GPT-NeoX because they lead to significant performance drop in oracle ICL-gold.

Method	Covered by $\mathcal{C}$							Not covered by $\mathcal{C}$			
	CR	Amz	Amz5	Yelp	Yelp5	Tweet	Avg	MR	SST2	SST5	Avg
Majority	$50.0_{0.0}$	$50.0_{0.0}$	$20.0_{0.0}$	$50.0_{0.0}$	$20.0_{0.0}$	$38.1_{0.0}$	$38.0_{0.0}$	$50.0_{0.0}$	$50.0_{0.0}$	$21.5_{0.0}$	$40.5_{0.0}$
Channel GPT-J											
No-demos	$73.2_{0.0}$	$86.1_{0.0}$	$34.4_{0.0}$	$88.0_{0.0}$	$36.6_{0.0}$	$\mathbf{47.6_{0.0}}$	$61.0_{0.0}$	$65.7_{0.0}$	$66.3_{0.0}$	$21.9_{0.0}$	$51.3_{0.0}$
Random inputs	$77.8_{2.4}$	$81.8_{3.2}$	$38.1_{1.6}$	$84.2_{4.6}$	$40.5_{1.4}$	$41.5_{1.1}$	$60.7_{2.4}$	$76.2_{3.6}$	$78.6_{3.6}$	$33.9_{3.6}$	$62.9_{3.6}$
Naive Z-ICL	$62.1_{0.8}$	$81.6_{0.5}$	$41.7_{0.4}$	$81.4_{0.3}$	$41.8_{0.8}$	$42.2_{1.0}$	$58.5_{0.6}$	$68.8_{0.4}$	$67.8_{0.8}$	$32.4_{0.6}$	$56.3_{0.6}$
Z-ICL (Ours)	$\mathbf{80.1_{0.1}}$	$\mathbf{88.9_{0.2}}$	$\mathbf{46.5_{0.4}}$	$\mathbf{88.4_{0.1}}$	$\mathbf{44.2_{0.3}}$	$46.8_{0.5}$	$\mathbf{65.8_{0.3}}$	$\mathbf{81.9_{0.1}}$	$\mathbf{82.6_{0.2}}$	$\mathbf{38.7_{0.5}}$	$\mathbf{67.7_{0.3}}$
ICL-gold (Oracle)	$84.4_{2.8}$	$90.9_{0.9}$	$45.5_{3.2}$	$91.0_{0.1}$	$47.4_{1.3}$	$48.0_{1.8}$	$67.9_{1.7}$	$86.9_{0.2}$	$88.8_{1.3}$	$42.1_{1.1}$	$72.6_{0.9}$
ICL-random (Oracle)	$82.3_{1.3}$	$91.3_{1.4}$	$44.9_{2.0}$	$91.1_{0.3}$	$48.0_{1.5}$	$46.8_{2.6}$	$67.4_{1.5}$	$86.6_{0.3}$	$86.1_{2.1}$	$41.8_{0.9}$	$71.5_{1.1}$
Direct GPT-J											
No-demos	$50.6_{0.0}$	$87.3_{0.0}$	$30.4_{0.0}$	$92.3_{0.0}$	$28.7_{0.0}$	$\mathbf{39.5_{0.0}}$	$54.8_{0.0}$	$51.7_{0.0}$	$52.9_{0.0}$	$26.8_{0.0}$	$43.8_{0.0}$
Random inputs	$71.1_{15.0}$	$91.2_{2.8}$	$37.5_{5.2}$	$91.5_{3.5}$	$36.4_{6.1}$	$28.8_{6.7}$	$59.4_{6.6}$	$68.2_{12.1}$	$69.9_{12.9}$	$30.1_{8.2}$	$56.1_{11.1}$
Naive Z-ICL	$65.2_{0.9}$	$89.3_{0.6}$	$\mathbf{39.6_{0.4}}$	$91.7_{0.6}$	$\mathbf{41.2_{0.8}}$	$32.3_{0.4}$	$59.9_{0.6}$	$64.6_{0.4}$	$66.1_{0.0}$	$30.9_{0.6}$	$53.9_{0.3}$
Z-ICL (Ours)	$\mathbf{78.8_{0.4}}$	$\mathbf{94.9_{0.1}}$	$38.5_{0.3}$	$\mathbf{96.0_{0.1}}$	$40.8_{0.3}$	$20.5_{0.1}$	$\mathbf{61.6_{0.3}}$	$\mathbf{81.0_{0.3}}$	$\mathbf{82.6_{0.2}}$	$\mathbf{30.9_{0.3}}$	$\mathbf{64.8_{0.3}}$
ICL-gold (Oracle)	$68.7_{13.9}$	$95.8_{0.1}$	$49.0_{3.8}$	$96.4_{0.4}$	$47.5_{5.8}$	$35.0_{5.1}$	$65.4_{4.9}$	$84.0_{6.8}$	$91.1_{3.2}$	$42.9_{0.9}$	$72.7_{4.0}$
ICL-random (Oracle)	$79.1_{10.0}$	$87.8_{7.5}$	$41.1_{4.8}$	$94.5_{1.9}$	$43.5_{3.5}$	$33.4_{2.7}$	$63.2_{5.1}$	$87.3_{3.6}$	$82.6_{9.7}$	$35.9_{3.5}$	$68.6_{5.6}$
Channel GPT-NeoX											
No-demos	$57.2_{0.0}$	$63.2_{0.0}$	$27.5_{0.0}$	$57.0_{0.0}$	$28.6_{0.0}$	$28.7_{0.0}$	$43.7_{0.0}$	$58.7_{0.0}$	$61.9_{0.0}$	$23.8_{0.0}$	$48.1_{0.0}$
Random inputs	$68.0_{4.2}$	$70.4_{2.3}$	$27.9_{1.9}$	$73.0_{3.1}$	$29.1_{1.9}$	$34.6_{4.9}$	$50.5_{3.1}$	$65.0_{4.9}$	$66.4_{5.2}$	$26.8_{3.6}$	$52.7_{4.6}$
Naive Z-ICL	$62.4_{0.2}$	$78.8_{0.9}$	$34.7_{1.2}$	$79.1_{0.8}$	$36.9_{0.8}$	$38.9_{0.5}$	$55.1_{0.7}$	$63.5_{0.8}$	$62.8_{0.7}$	$29.9_{0.8}$	$55.1_{0.7}$
Z-ICL (Ours)	$\mathbf{79.0_{0.2}}$	$\mathbf{84.3_{0.7}}$	$\mathbf{37.8_{0.5}}$	$\mathbf{87.0_{0.4}}$	$\mathbf{39.9_{1.0}}$	$\mathbf{46.7_{0.6}}$	$\mathbf{62.5_{0.6}}$	$\mathbf{73.2_{0.3}}$	$\mathbf{74.3_{0.2}}$	$\mathbf{33.2_{0.3}}$	$\mathbf{60.2_{0.3}}$
ICL-gold (Oracle)	$85.5_{2.3}$	$90.3_{0.8}$	$41.6_{1.8}$	$86.8_{2.8}$	$43.5_{0.7}$	$47.9_{1.9}$	$65.9_{1.7}$	$86.2_{0.8}$	$89.4_{0.9}$	$40.8_{1.1}$	$72.1_{0.9}$
ICL-random (Oracle)	$78.1_{3.3}$	$88.5_{1.5}$	$39.8_{1.4}$	$88.0_{1.7}$	$43.5_{1.6}$	$44.0_{1.1}$	$63.7_{1.8}$	$86.3_{0.9}$	$88.1_{1.6}$	$39.9_{1.2}$	$71.4_{1.2}$
Direct GPT-NeoX											
No-demos	$61.5_{0.0}$	$50.8_{0.0}$	$20.2_{0.0}$	$72.2_{0.0}$	$21.3_{0.0}$	$30.8_{0.0}$	$42.8_{0.0}$	$49.9_{0.0}$	$49.1_{0.0}$	$17.5_{0.0}$	$38.8_{0.0}$
Random inputs	$72.5_{13.7}$	$83.5_{12.9}$	$38.7_{3.6}$	$85.0_{8.4}$	$37.1_{2.6}$	$36.4_{9.5}$	$58.9_{8.5}$	$74.9_{8.7}$	$78.2_{9.4}$	$37.5_{6.2}$	$63.5_{8.1}$
Naive Z-ICL	$76.2_{0.3}$	$87.5_{0.7}$	$41.2_{0.9}$	$89.0_{0.8}$	$\mathbf{39.1_{0.6}}$	$\mathbf{40.2_{0.9}}$	$62.2_{0.7}$	$71.7_{1.1}$	$73.8_{1.0}$	$\mathbf{34.0_{0.5}}$	$59.8_{0.9}$
Z-ICL (Ours)	$\mathbf{91.4_{0.3}}$	$\mathbf{94.0_{0.1}}$	$\mathbf{41.2_{0.4}}$	$\mathbf{92.2_{0.3}}$	$38.6_{0.3}$	$35.2_{0.9}$	$\mathbf{65.4_{0.4}}$	$\mathbf{84.0_{0.4}}$	$\mathbf{87.8_{0.7}}$	$33.3_{0.6}$	$\mathbf{68.4_{0.6}}$
ICL-gold (Oracle)	$78.5_{14.8}$	$95.6_{0.5}$	$47.0_{2.7}$	$91.7_{3.6}$	$40.6_{3.1}$	$32.8_{6.5}$	$64.4_{5.2}$	$89.0_{0.9}$	$88.6_{5.1}$	$43.0_{3.1}$	$73.5_{3.0}$
ICL-random (Oracle)	$78.5_{13.6}$	$92.9_{2.5}$	$45.6_{1.6}$	$88.5_{4.3}$	$41.3_{3.5}$	$33.1_{3.9}$	$63.3_{4.9}$	$81.2_{13.7}$	$76.9_{13.8}$	$37.5_{3.1}$	$65.2_{10.2}$

Table 6: Results with GPT-J and GPT-NeoX. *Oracle* indicates the method has access to the training data, thus is not comparable with the rest of the models. Covered/not covered by $\mathcal{C}$ indicates whether or not the domain of the dataset is covered by our text corpus. Z-ICL is significantly better than previous zero-shot (No-demos) on all datasets, and is on par with ICL-gold on datasets covered by $\mathcal{C}$.

Method	Covered by $\mathcal{C}$					Not covered by $\mathcal{C}$
	CR	Amz	Yelp	Tweet	Avg.	SST-2
Majority	$50.0_{0.0}$	$50.0_{0.0}$	$50.0_{0.0}$	$38.1_{0.0}$	$47.6_{0.0}$	$50.0_{0.0}$
Channel GPT-3						
No-demos	$76.6_{0.0}$	$77.2_{0.0}$	$\mathbf{88.0_{0.0}}$	$36.2_{0.0}$	$69.5_{0.0}$	$80.8_{0.0}$
Z-ICL (Ours)	$\mathbf{80.8_{0.6}}$	$\mathbf{89.1_{0.3}}$	$87.6_{0.0}$	$\mathbf{41.4_{0.4}}$	$\mathbf{73.4_{0.6}}$	$\mathbf{82.4_{74.7}}$
ICL-gold (Oracle)	$74.2_{7.4}$	$86.0_{3.6}$	$91.7_{0.9}$	$43.8_{0.2}$	$73.9_{3.0}$	$88.1_{1.1}$
ICL-random (Oracle)	$73.9_{3.9}$	$83.4_{4.8}$	$90.4_{1.4}$	$41.4_{2.0}$	$72.3_{3.0}$	$84.8_{1.2}$
Direct GPT-3						
No-demos	$68.4_{0.0}$	$88.2_{0.0}$	$96.4_{0.0}$	$\mathbf{37.8_{0.0}}$	$72.7_{0.0}$	$73.2_{0.0}$
Z-ICL (Ours)	$\mathbf{71.9_{0.1}}$	$\mathbf{93.0_{0.2}}$	$\mathbf{97.7_{0.3}}$	$28.3_{0.4}$	$\mathbf{72.7}_{0.3}$	$\mathbf{78.1_{0.1}}$
ICL-gold (Oracle)	$79.5_{9.5}$	$97.0_{0.2}$	$98.5_{0.1}$	$30.5_{8.0}$	$79.3_{2.5}$	$94.2_{0.2}$
ICL-random (Oracle)	$81.0_{6.8}$	$95.4_{0.6}$	$93.7_{2.1}$	$42.2_{39.4}$	$77.4_{2.7}$	$93.9_{0.5}$

Table 7: Results on GPT-3 on a subset of evaluation datasets. *Oracle* indicates the method has access to the training data, thus is not comparable with the rest of the model. Covered/not covered by $\mathcal{C}$ indicates whether or not the domain of the dataset is covered by our text corpus. Z-ICL is consistently better than the previous zero-shot (No-demos) on all datasets, even with a template.

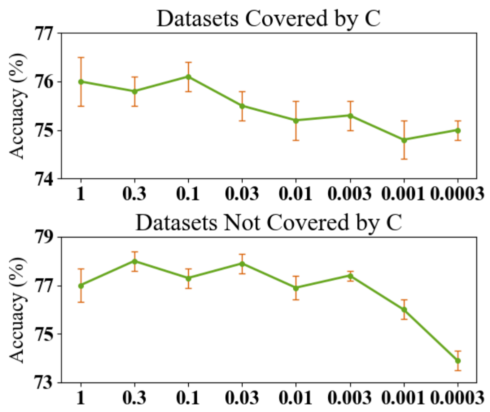
C.3 Additional Ablations

We perform additional ablation studies in complement to Section 4.2.

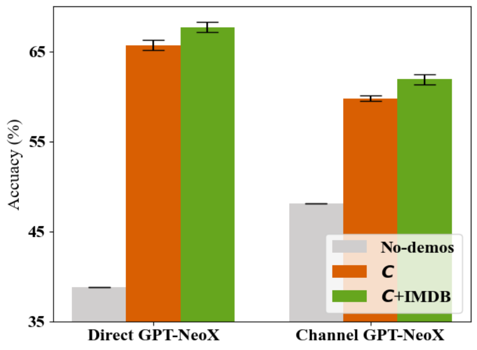
Effect of the size of the corpus. We quantify the impact of the size of the corpus. This is important to judge whether Z-ICL can potentially achieve better results by scaling the corpus. We evaluate Z-ICL with a corpus with varying sizes, from 100% to 0.03% of the corpus.

	GPT-J	GPT-NeoX
Total	82.3	88.0
Correct	90.8	94.2
Incorrect	73.9	81.7

Table 8: % of predictions that match the label of the demonstration example that is identical to the test input. Evaluated on CR with channel GPT-J and channel GPT-NeoX. The model copies the label paired with an identical example in the majority of cases.



(a) **Effect of the size of the corpus.** The x-axis indicates the size of the corpus, varying from 160M paragraphs (1) to 48K paragraphs (0.0003). Performance goes down as the corpus size decreases.



(b) **Effect of the coverage of the corpus.** Performance of Z-ICL before and after IMDB is added to the corpus. Expanding the coverage of the corpus consistently improves the performance despite only 2% of the increase in the size of the corpus.

Figure 3

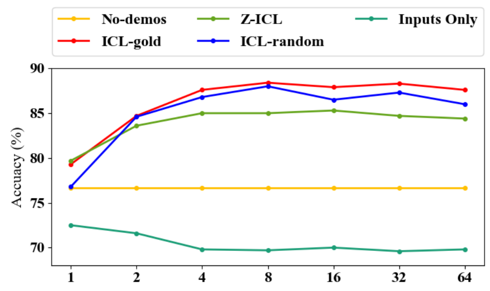
Figure 3a demonstrates that performance goes down as the size of the corpus gets smaller. This is likely because there are less sentences that are sufficiently close to the test input when the corpus is smaller, thus the *relevance* of the nearest neighbors and the test input drops. This trend is clearer on the datasets covered by $\mathcal{C}$ than on the datasets not covered by $\mathcal{C}$.

Effect of the format of demonstrations. How many input-label pairs does Z-ICL need to benefit from pseudo-demonstrations? Are gains from pseudo-demonstrations mainly from the fact that the LM conditions on relevant text, or does the LM benefit from a specific format of the pseudo-demonstrations: a concatenation of input-label pairs? To answer these questions, we experiment with (1) Z-ICL with varying range of k from 1 to 64, and (2) a variant of Z-ICL where the LM conditions on a concatenation of retrieved inputs, without randomly paired labels (called "Inputs-only").

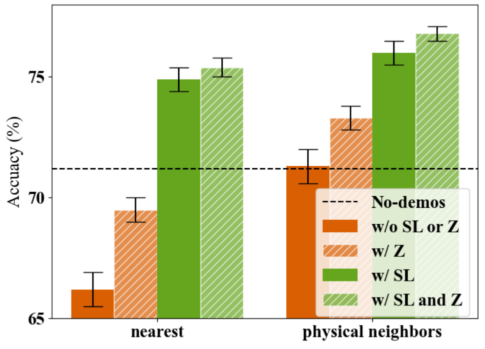
Results are shown in Figure 4a. First, Z-ICL is significantly better than zero-shot baselines and stays on par with the oracle baselines consistently across different values of k. Moreover, using no labels ("Inputs-only") performs significant worse than its counterparts. This suggests that Z-ICL takes advantages of the form of input-label pairs, and is beyond simply conditioning on relevant context.

Effect of the coverage of the corpus. We quantify the impact of the coverage of the corpus, and whether adding more domains in the corpus improves performance. We do so by adding the unlabeled portion of IMDB review (Maas et al., 2011) to the corpus $\mathcal{C}$. The size of $\mathcal{C}$ increases only by 2%, but covers the domain of three datasets that were previously not covered (SST2, SST5 and MR).

Figure 3b shows the performance on three datasets before and after adding the IMDB corpus. Performance improves consistently over all LMs, even though it only adds up the size by 2%. This suggests that the coverage of the text corpus is important, and it is feasible to further improve the overall performance simply by expanding the corpus.



(a) **Effect of the format of demonstrations** with varying numbers of demonstrations (k). Z-ICL consistently performs on par with the oracle baseline, and "Inputs-only" performs significantly worse.



(b) **Quantifying the Copying Effect.** *SL and* Z stand for synonym labeling and zeroing-out the induction heads, respectively. Techniques designed for reducing the copying effect (physical neighbor and synonym labeling) see less effect from zeroing out the induction heads.

Figure 4

D Limitations

Extension to multi-sentence tasks. Our experiments are limited to single-sentence tasks, as we only retrieve single-sentence nearest neighbors to a test input. Multi-sentence tasks such as natural language inference would require constructing pseudo-demonstrations that consists of multiple sentences, which we leave for future work.

Beyond classification. Our experiments are limited to classification. Extensions to multi-choice tasks or generation tasks are not trivial, because there is no fixed set of options that are shared between inputs in the demonstrations and the test input. We leave extensions to non-classification tasks for future work.

Better construction of pseudo-demonstrations. We think future work can explore better constructing the pseudo-demonstrations. For instance, this paper uses manually chosen synonym labels (details in Appendix B.3). We hypothesize that better pseudo-demonstrations can improve performance, which we leave for future work.