Scaling Laws for Many-Shot In-Context Learning with Self-Generated Annotations

Anonymous Authors[1]

Abstract

The high cost of obtaining high-quality annotated data for in-context learning (ICL) has motivated the use of self-generated annotations as a substitute for ground-truth labels. While such methods have shown promise in few-shot settings, their effectiveness in many-shot scenarios remains underexplored. To address this gap, we propose a simple baseline, Naive-SemiICL, which follows a three-step framework—annotation generation, demonstration selection, and in-context inference—and demonstrates clear scaling trends across both discriminative and generative tasks. Naive-SemiICL outperforms few-shot ICL at various ground truth data budgets, notably surpassing 16-shot baselines by 9.94% across 16 tasks on GPT-4o-mini. We further introduce IterPSD, an annotation method that iteratively improves pseudo-annotation quality by augmenting its prompt with self-annotated examples. IterPSD yields additional 6.8% gains on 5 classification tasks compared to Naive-SemiICL. Code is available at: `https://anonymous.4open.science/r/semi-supervised-icl-FA07`

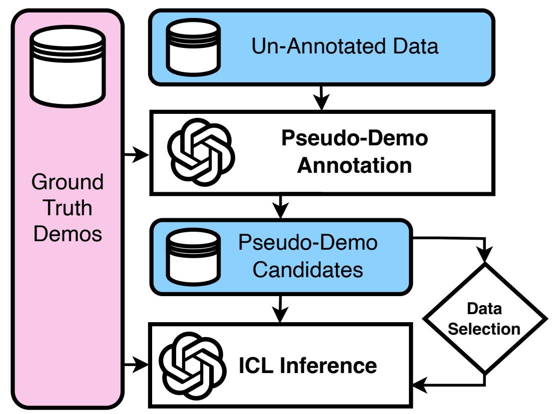

Figure 1: Semi-supervised ICL Framework. Ground truth data are used as demonstration for generating pseudo-demonstrations from unannotated data. The generated pseudo-demonstrations conjunctively with a small ground truth demonstration, are selectively used as demonstrations for the final prompting.

1. Introduction

In-context learning (ICL) has emerged as a powerful paradigm in natural language processing, enabling language models (LMs) to learn, adapt, and generalize from examples presented within their input context. This approach eliminates the need for extensive retraining and parameter modifications, facilitating more flexible and efficient learning (Brown et al., 2020; Min et al., 2022; Agarwal et al., 2024; Fang et al., 2025). The high cost of obtaining high-quality annotated data for ICL has motivated the development of methods (Zhang et al., 2023; Li & Qiu, 2023; Mamooler et al., 2024; Li et al., 2024a; Chen et al., 2023) that use self-generated annotations in place of ground-truth labels. However, previous research has not examined ICL performance with self-generated annotations in *many-shot settings*. Recently, (Agarwal et al., 2024) established a scaling law, showing that ICL performance improves with the number of demonstrations—up to thousands of examples. Inspired by this finding, we pose the following question:

> Research Question:
>
> *Can we scale ICL performance using self-generated demonstrations up to thousands of examples as well?*

We systematically investigate this question under a three-step framework (Figure 1): ① annotation generation, ② demonstration selection, and ③ semi-supervised inference, which we term *Semi-Supervised ICL*. We first introduce a simple baseline, Naive-SemiICL, which annotates unlabeled data in a single iteration, scoring each annotation using the LLM's verbalized confidence. Naive-SemiICL consistently outperforms ICL baselines in zero-shot, few-shot, and many-shot settings, as well as prior methods. We highlight that Naive-SemiICL achieves optimal performance with ***1000***

demonstrations on certain tasks (Figure 2).

With potentially thousands of self-annotated examples in the prompt, each demonstration can be viewed as a *dataset*, which motivates the following question:

> Research Question:
>
> *In what ways can techniques from traditional semi-supervised learning be leveraged to improve ICL performance?*

We address this question by proposing *IterPSD*, an iterative approach that progressively refines pseudo-demonstration quality by incorporating self-generated annotations at each iteration. IterPSD further improves semi-supervised ICL performance on five classification tasks, achieving gains of up to 6.8% (Table 1).

2. Method

2.1. Semi-Supervised ICL

Semi-supervised ICL is a three-step framework consisting of ① pseudo-demonstration generation, ② pseudo-demonstration selection, and ③ in-context inference. During step ①, Semi-supervised ICL annotates large set of *unannotated data* $\mathcal{X}_u = \{x_i\}^{N_u}$, using a small set of *ground-truth data* $\mathcal{E}_g = \{(x_i, y_i)\}^{N_l}$ (or none) as demonstrations. For each annotation, we generate a confidence measure c along with the prediction y by conditioning on a prompt ρ, a set of demonstrations $\mathcal{E}$, and an input x.

$$y, c = \mathrm{LLM}(\rho, \mathcal{E}, x) \tag{1}$$

We define the prediction y broadly here. y could be labels in a classification task, a short paragraph in a question answering task, or a reasoning chain that includes the final answer in a reasoning task. We denote the resulting set of annotations as

$$\mathcal{D}_{\mathrm{PSD}} = \{(x, y, c) | x \in \mathcal{X}_u\}, \tag{2}$$

where y and c are generated from Equation 1.

We then sample pseudo-demonstrations from annotations whose confidence surpasses some threshold $c \geq \lambda$.

$$\mathcal{E}_u = \mathrm{Sampler}(\mathcal{D}_{\mathrm{PSD}}, \lambda) \tag{3}$$

During inference, we prompt the LLM with both sampled pseudo-demonstrations and the ground-truth data used to annotate them.

$$y = \mathrm{LLM}(\rho, \mathcal{E}_u \cup \mathcal{E}_g, x) \tag{4}$$

Algorithm 1 IterPSD

Input: prompt ρ, ground-truth demonstrations $\mathcal{E}_g$, chunk size K, ratio of random examples ϵ, maximum number of pseudo-demonstrations κ, sampler for unannotated data $\mathrm{Sampler}_u$, sampler for pseudo-demonstrations $\mathrm{Sampler}_{\mathrm{PSD}}$;
Initialize $\mathcal{D}_{\mathrm{PSD}} = \emptyset$; {*Set of all the annotated pseudo-demonstrations.*}
Initialize $\overline{\mathcal{D}}_{\mathrm{PSD}} = \mathcal{X}_u$; {*Set of data yet to be annotated.*}
Initialize $\mathcal{E} = \mathcal{E}_g$; {*Demonstration for generating pseudo-demonstrations.*}
while $\overline{\mathcal{D}}_{\mathrm{PSD}} \neq \emptyset$ **do**
  if $|\mathcal{E}| > \kappa$ **then**
    $\mathcal{E}$ = top-κ confident examples in $\mathcal{D}_{\mathrm{PSD}}$;
    {*Cap the demonstration at a maximum size*}
  end if
  $S_u = \mathrm{Sampler}_\epsilon(\mathcal{D}_{\mathrm{PSD}}, \overline{\mathcal{D}}_{\mathrm{PSD}}, K, \epsilon)$; {*Retrieves a sample of size K using ϵ-Random Sampler*}
  S_{PSD} = Naive-SemiICL$(S, \rho, \mathcal{E}_g \cup \mathcal{D}^{\lambda}_{\mathrm{PSD}})$;
  {*One iteration of Naive-SemiICL.*}
  $S^{\lambda}_{\mathrm{PSD}} = \mathrm{Filter}(S_{\mathrm{PSD}}, \lambda)$;
  $\mathcal{E} = \mathcal{E} \cup S^{\lambda}_{\mathrm{PSD}}$;
  $\mathcal{D}_{\mathrm{PSD}} = \mathcal{D}_{\mathrm{PSD}} \cup S_{\mathrm{PSD}}$;
  $\overline{\mathcal{D}}_{\mathrm{PSD}} = \overline{\mathcal{D}}_{\mathrm{PSD}} - S_{\mathrm{PSD}}$;
end while
Return $\mathcal{D}_{\mathrm{PSD}}$;

2.2. A Simple Baseline for Semi-Supervised ICL

We propose a simple method, Naive-SemiICL, that generates pseudo-demonstrations in a single iteration. Naive-SemiICL generates a prediction y and a confidence score c for each unlabeled instance by going through unannotated data exactly once. As a basic form of Semi-Supervised ICL, Naive-SemiICL's effectiveness relies on the successful filtering of low-quality annotations. We detail the Naive-SemiICL in Algorithm 2.

2.3. Iterative Pseudo-Demonstration Generation

As we will show in Section 4, ICL inference accuracy begins to improve with a relatively small amount of pseudo-demonstrations well below the amount that achieves the optimal performance. One could improve the quality of subsequent annotations by incorporating self-annotated examples as demonstrations during pseudo-annotation. Motivated by this insight, we design IterPSD (Algorithm 1). In each iteration, IterPSD samples and annotates K pseudo-demonstrations. Pseudo-demonstrations with a confidence higher than λ are added to the existing set of demonstrations used for pseudo-annotation. When the number of demonstrations in the prompt reaches an upper limit κ, we resample the κ most confident examples from all previously

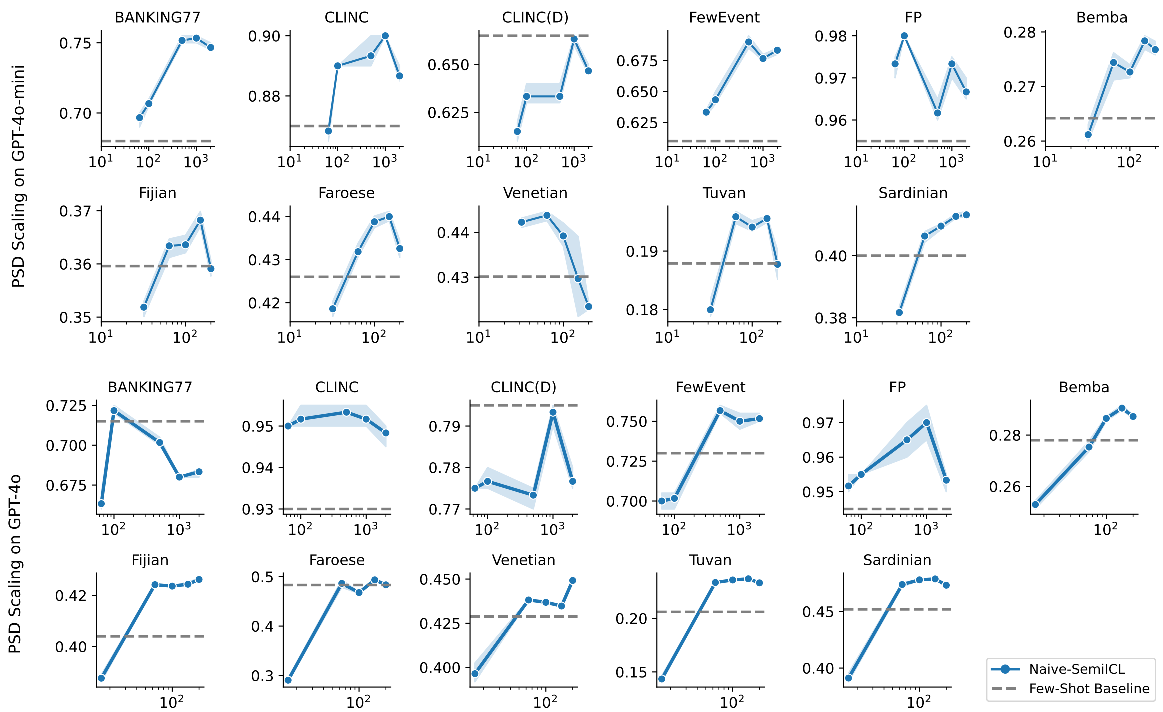

Figure 2: Scaling trend of Naive-SemiICL (Verbalized Confidence) on classification and translation tasks with GPT-4o and GPT-4o-mini. The dashed gray line represents the few-shot baseline. Both model exhibits a scaling trend on most tasks.

annotated examples whose confidence score is higher than λ. We offer additional empirical motivations for IterPSD in Appendix F.1.

Curriculum Learning. In curriculum learning (Soviany et al., 2021), training examples are organized and presented in increasing order of difficulty to facilitate more effective learning. We adapt this idea to ICL by sampling unannotated examples that are similar to those already annotated, thereby introducing harder examples progressively. However, we find that sampling only similar examples introduces a strong bias toward examples annotated later in the process. Thus, we design the sampler for unannotated data to retrieve both similar and diverse data. The ϵ-Random Sampler (Algorithm 3) selects $(1-\epsilon)$ of the examples to be similar and randomly samples the rest. We compute the similarity between an annotated example and an unannotated example using the text embedding of their problem statement x. Since it has been shown that simply seeing similar examples could boost in-context prediction accuracy of LLMs (Min et al., 2022), the random portion of the sample ensures that subsequent annotations are covered by previously seen examples. In practice, we find that sampling 80% of each batch randomly yields the best performance.

3. Experimetnal Setup

Benchmark. Our evaluation covers 16 datasets spanning classification, translation, and reasoning tasks. Detailed descriptions of the datasets are provided in Appendix A, and the prompts used for each task are summarized in Table 4.

Evaluation Metrics. For all classification and reasoning tasks, we report **accuracy** as the performance metric. For translation tasks, following (Agarwal et al., 2024), we report the **ChrF++** score (Popović, 2015) using the default configuration from TorchMetrics (Detlefsen et al., 2022).

Baselines & Configurations. We compare Naive-SemiICL to k-shot ICL, ensuring both methods use the same amount of ground-truth data. To assess the role of confidence-based data selection, we include an unfiltered variant of Naive-SemiICL that samples pseudo-annotations without applying the filtering step. We also compare Naive-SemiICL to MoT (Li & Qiu, 2023), a method tailored to reasoning tasks; all comparisons with MoT are conducted using 16 ground-truth examples. For IterPSD, we use the same ground-truth budget (16 examples) and compare its performance to Naive-SemiICL under the identical budget. Hyperparameter settings are detailed in Appendix B.

Confidence Scores. We primarily report results using Verbalized Confidence, where the LLM is prompted to generate

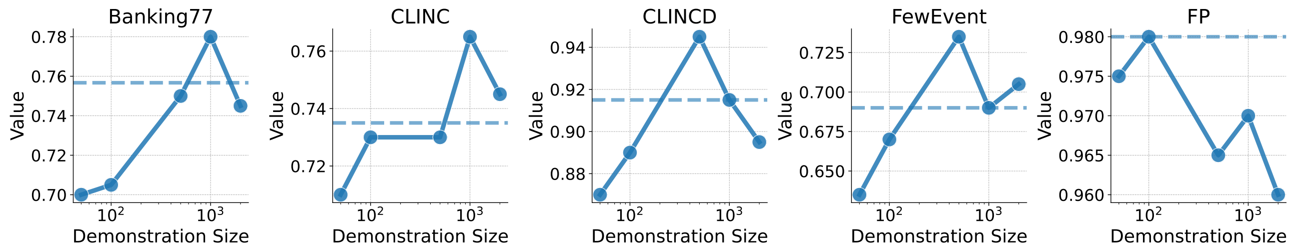

Figure 3: Scaling trend of IterPSD (Verbalized Confidence) on five benchmark tasks. Blue horizontal dashed line represents the best performing Naive-SemiICL on the same dataset.

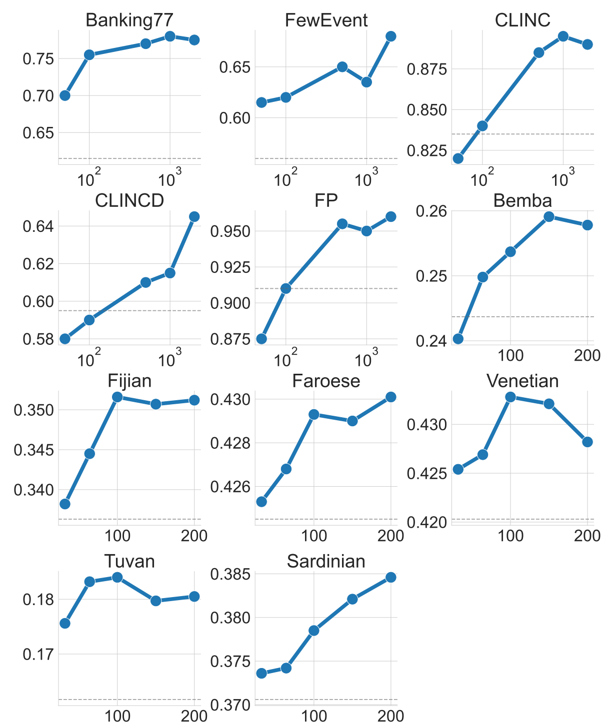

Figure 4: Scaling trend of Naive-SemiICL with no initial ground truth data. Grey dash line represents the prediction performance of zero-shot prompting. All results obtained from GPT-4o-mini

a confidence score for each prediction. For IterPSD, we also evaluate Self-Consistency, which estimates confidence by sampling nn predictions and using the majority vote frequency. Additional details are provided in Appendix D.

Models. All experiments are conducted using GPT-4o-mini and GPT-4o, checkpointed on 2024-07-18 and 2024-11-20, respectively. We discuss the computational cost of our experiments in Appendix C.

4. Experimental Results

Figure 2 illustrates the scaling behavior of Naive-SemiICL as the number of pseudo-demonstrations increases, using 16 ground-truth examples. Naive-SemiICL consistently matches or outperforms the baseline across all tasks, exhibiting a clear scaling trend. For classification tasks, peak performance typically occurs between 500 and 1000 pseudo-demonstrations, while for translation tasks, it is reached between 100 and 200. A comparison of Naive-SemiICL under different confidence scoring methods is provided in Appendix F.

A similar pattern is observed for IterPSD (Figure 3), which also achieves optimal performance between 500 and 1000 pseudo-demonstrations on classification tasks. A more detailed comparison between IterPSD and Naive-SemiICL is included in Appendix E.2.

Importantly, the scaling trend of Naive-SemiICL remains consistent across different ground-truth budgets. Figure 4 shows its performance in the zero-shot setting, where it outperforms the baseline on all tasks, achieving an average gain of 11.36% under GPT-4o-mini. This exceeds the 9.94% improvement observed in the 16-shot setting, highlighting Naive-SemiICL's effectiveness in resource-constrained scenarios. Additional results under many-shot ground-truth settings are presented in Appendix E.1.

5. Conclusion

By observing empirical scaling trends with both Naive-SemiICL and IterPSD, we demonstrate that in-context learning performance can be scaled using thousands of self-generated pseudo-demonstrations. We further highlight the versatility of Naive-SemiICL, showing that it consistently outperforms k-shot ICL across a wide range of ground-truth data budgets. Finally, we validate the framework of semi-supervised in-context learning by incorporating curriculum learning principles into the design of IterPSD, which achieves superior performance over Naive-SemiICL on classification tasks.

References

Agarwal, R., Singh, A., Zhang, L. M., Bohnet, B., Rosias, L., Chan, S. C., Zhang, B., Anand, A., Abbas, Z., Nova, A., Co-Reyes, J. D., Chu, E., Behbahani, F., Faust, A., and Larochelle, H. Many-shot in-context learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=AB6XpMzvqH.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Casanueva, I., Temcinas, T., Gerz, D., Henderson, M., and Vulic, I. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*, mar 2020. URL https://arxiv.org/abs/2003.04807. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Chen, W., Wu, C., Chen, Y., and Chen, H. Self-icl: Zero-shot in-context learning with self-generated demonstrations. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15651–15662. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.968. URL https://doi.org/10.18653/v1/2023.emnlp-main.968.

Costa-Jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Deng, S., Zhang, N., Kang, J., Zhang, Y., Zhang, W., and Chen, H. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 151–159, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371796. URL https://doi.org/10.1145/3336191.3371796.

Detlefsen, N. S., Borovec, J., Schock, J., Jha, A. H., Koker, T., Di Liello, L., Stancl, D., Quan, C., Grechkin, M., and Falcon, W. Torchmetrics-measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022.

Fang, L., Liu, A., Zhang, H., Zou, H. P., Zhang, W., and Yu, P. S. Tabgen-icl: Residual-aware in-context example selection for tabular data generation. *arXiv preprint arXiv:2502.16414*, 2025.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., and Fan, A. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzano, M. A., Tang, L., and Mars, J. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL https://www.aclweb.org/anthology/D19-1131.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Li, X. and Qiu, X. MoT: Memory-of-thought enables ChatGPT to self-improve. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6354–6374, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.392. URL https://aclanthology.org/2023.emnlp-main.392/.

Li, Y., Korhonen, A., and Vulić, I. Self-augmented in-context learning for unsupervised word translation. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp.

743–753, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.67. URL https://aclanthology.org/2024.acl-short.67/.

Li, Y., Korhonen, A., and Vulić, I. Self-augmented in-context learning for unsupervised word translation. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 743–753, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.67. URL https://aclanthology.org/2024.acl-short.67/.

Malo, P., Sinha, A., Korhonen, P. J., Wallenius, J., and Takala, P. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2013. URL https://api.semanticscholar.org/CorpusID:7700237.

Mamooler, S., Montariol, S., Mathis, A., and Bosselut, A. Picle: Pseudo-annotations for in-context learning in low-resource named entity detection. *CoRR*, abs/2412.11923, 2024. doi: 10.48550/ARXIV.2412.11923. URL https://doi.org/10.48550/arXiv.2412.11923.

McLachlan, G. J. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Popović, M. chrF: character n-gram F-score for automatic MT evaluation. In Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Hokamp, C., Huck, M., Logacheva, V., and Pecina, P. (eds.), *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049/.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948. URL http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 596–608, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/06964dce9addb1c5cb5d6e3d9838f733-Paper.pdf.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. Curriculum learning: A survey. *CoRR*, abs/2101.10382, 2021. URL https://arxiv.org/abs/2101.10382.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., , and Wei, J. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

Xiong, M., Hu, Z., Lu, X., LI, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Zhang, Z., Zhang, A., Li, M., and Smola, A. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=5NTt8GFjUHkr.

Zou, H. and Caragea, C. JointMatch: A unified approach for diverse and collaborative pseudo-labeling to semi-supervised text classification. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7290–7301, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.451. URL https://aclanthology.org/2023.emnlp-main.451/.

Zou, H., Zhou, Y., Zhang, W., and Caragea, C. DeCrisisMB: Debiased semi-supervised learning for crisis tweet classification via memory bank. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6104–6115, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.406. URL https://aclanthology.org/2023.findings-emnlp.406/.

Zou, H. P., Caragea, C., Zhou, Y., and Caragea, D. Semi-supervised few-shot learning for fine-grained disaster tweet classification. In *Proceedings of the 20th International ISCRAM Conference*. ISCRAM 2023, 2023b.

Zou, H. P., Gu, Z., Zhou, Y., Chen, Y., Zhang, W., Fang, L., Wang, Y., Li, Y., Liu, K., and Yu, P. S. Testnuc: Enhancing test-time computing approaches through neighboring unlabeled data consistency. *arXiv preprint arXiv:2502.19163*, 2025a.

Zou, H. P., Singh, S., Nian, Y., He, J., Cai, J., Mansour, S., and Su, H. Glean: Generalized category discovery with diverse and quality-enhanced llm feedback. *arXiv preprint arXiv:2502.18414*, 2025b.

A. Datasets

Classification Datasets.

- **BANKING77.** The BANKING77(Casanueva et al., 2020) dataset is a fine-grained intent classification benchmark in the banking domain, consisting of 13,083 customer queries labeled into 77 intent categories.
- **CLINC.** The CLINC150 (Larson et al., 2019) dataset is a benchmark for intent classification, containing 22,500 user queries across 150 intent categories grouped into 10 domains, along with an out-of-scope category. We refer to the intent classification task of CLINC150 as CLINC.
- **CLINC(D).** We refer to the domain classification annotation of CLINC150 as CLINC(D).
- **FewEvent.** The FewEvent(Deng et al., 2020) dataset contains 4,436 event mentions across 100 event types, with each event type having only a few annotated examples (typically 5 to 10 per type).
- **FP.** Financial Phrasebank(Malo et al., 2013) The Financial PhraseBank dataset consists of 4840 sentences from English language financial news categorised by sentiment.

Low-Resource Language Translation. FLORES-200 (Costa-Jussà et al., 2022) contains 200 languages translated from a common corpus. It is an extension of the original FLORES-101 (Goyal et al., 2022) dataset, which covered 101 languages. The dataset covers low-resource and high-resource languages, including many languages with little prior data on. It includes many African, South Asian, and Indigenous languages, making it one of the most diverse multilingual benchmarks.

Reasoning Datasets.

- **GPQA.** GPQA(Rein et al., 2024) is a multiple-choice question answering benchmark, with graduate-level questions that involves reasoning in biology, physics, and chemistry.
- **LiveBench Math.** LiveBenchMath contains 368 contamination-free mathematical problems, sampled from high school math competitions, proof-based fill-in-the-blank questions from Olympiad-level problems, and an enhanced version of the AMPS dataset.
- **BigBenchHard.** We include three tasks from BigBenchHard(Suzgun et al., 2022). **Logical7** evaluates a model's ability to deduce the order of a sequence of objects based on provided clues about their spatial relationships and placements. The **Geometric Shapes** task within the BigBenchHard evaluates a model's ability to interpret and identify geometric figures based on SVG path data. The **Date** task within the BigBenchHard benchmark evaluates a model's ability to comprehend and manipulate date-related information.

A.1. Train-Test Split

For classification tasks with more than 5,000 examples, we randomly sample 5,000 examples for demonstration and 200 for evaluation. For tasks with less than 5,000 examples, we randomly sample 200 for evaluation and use the rest for demonstration. Each FLORES dataset is comprised of a development set with 997 examples and a development test set with 1012 examples. We use all of 997 for demonstration and randomly sample 200 from the development test examples for evaluation. We use the diamond split (198 examples) of GPQA following (Agarwal et al., 2024), out of which 99 are used for evaluation and the other 99 are used for demonstration. Since LiveBench Math contains math problems from three sources, we evenly sample 150 questions from different sources for evaluation and use the rest for demonstration. Each BigBenchHard dataset contains 250 examples. We randomly sample 100 for evaluation and use the rest for prompting.

B. Hyperparameters

Confidence Thresholds. Unless stated otherwise, we filter all generated pseudo-demonstrations using the confidence threshold at the 90th percentile.

IterPSD. We found the optimal chunk size K to be 500 on most tasks except on FP. We uses $\epsilon = 0.8$ on all tasks. We also find that $\kappa = 1000$ yielded the best results on all tasks except on FP where the optimal $\kappa = 300$.

MoT. Following (Li & Qiu, 2023), we use 5 clusters for demonstration retrieval. Like IterPSD, we use OpenAI's *text-embedding-3-large* for similarity-based retrieval. We set the confidence threshold to 90-th percentile for the entropy-based filtering.

C. Computational Budget

All experiments were conducted on an Apple M3 chip. During IterPSD, embedding-based retrieval accounted for less than 1% of the total computation time. Embeddings were retrieved from the OpenAI API at a latency of approximately 400ms per example and can be precomputed during dataset preprocessing, as each embedding needs to be computed only once. The cost of generating embeddings is 0.13 per million tokens. All experiments were completed within a $1,000$ budget.

D. Confidence Metrics

We primarily evaluate three confidence metrics: Verbalized Confidence, Entropy, and Self-Consistency, as defined below. We also experimented with Back-Translation for translation tasks.

Verbalized Confidence. Verbalized Confidence (Xiong et al., 2024) prompts the LLM to generate the confidence score as part of its response. See Table 4 for how we induce the Verbalized Confidence scores from LLMs in prompts.

Entropy. Entropy (Shannon, 1948) estimates the uncertainty of generated content using the token probability $P(w_i \mid w_{<i})$, where w_i denotes the generated tokens and $w_{<i}$ represents the preceding tokens in the prompt:

$$c_{\text{Ent}} = -\frac{1}{L}\sum_{i=s}^{L} \log P(w_i \mid w_{<i}). \tag{5}$$

We find Entropy unsuitable for estimating uncertainty on classification tasks, as it predominantly returns a confidence score of one, making the data selection step redundant.

Self-Consistency. Self-consistency (Wang et al., 2023) samples multiple responses using diverse decoding paths and selects the most consistent answer based on majority voting. The relative frequency of the majority answer y^{maj} naturally defines a confidence score for the generated annotations. Let K_i be the size of the equivalence class $y_i \subseteq \{\tilde{y}_i\}$. Then the Self-Consistency Confidence equals

$$c_{\text{SC}} = \max_i \frac{K_i}{K}. \tag{6}$$

Back-Translation. Suppose an LLM has translated a source language input s into a target language output t. We then use the same LLM to translate t back to the original language

$$\hat{s} = \text{LM}(t, \rho_b),$$

where ρ_b is a prompt that induces the back-translation. Then, the Back-Translation Confidence is the cosine similarity between the original input s and the back-translation $\hat{s}$

$$c = \text{sim}_{\cos}(\phi(\hat{s}), \phi(s)),$$

where ϕ is an embedding function.

E. Extended Experiments

E.1. Naive-SemiICL with Expanded Ground Truth Budget

we found Naive-SemiICL to be effective in high-resource settings. Figure 5 compares the performance of Naive-SemiICL and ground-truth ICL when $k_l \in \{64, 100, 500\}$ ground-truth examples are available. Across three tasks, Naive-SemiICL consistently outperforms the corresponding k-shot baselines. We observe diminishing returns in performance gains as the number of annotated demonstrations increases. On average, $k_g = 64$ improves performance by 10.49% over the baseline, whereas $k_g = 500$ yields only a 4.73% improvement across the three tasks. Combining these results, Naive-SemiICL is most effective when ground-truth data is scarce, although it can still be effective in high-resource settings.

Method	BANKING	CLINC	CLINC(D)	FewEvent	FP
Naive-V	75.67	69.00	90.00	66.50	98.00
Naive-S	75.00	73.50	91.50	69.00	98.00
Iter-V	**78.00**	69.00	90.50	**73.50**	98.00
Iter-S	**78.00**	**78.50**	**94.50**	70.00	**98.50**
Improvement	**3.10%**	**6.80%**	**3.28%**	**6.52%**	**0.50%**

Table 1: Comparison of Naive-SemiICL (Naive) and IterPSD (Iter) methods on various datasets using GPT-4o-mini, evaluated using verbalized (-V) and self-consistency (-S) confidence scores. The best-performing results for each dataset are highlighted in bold, while the second-best results are underlined.

E.2. Extended Experiments on IterPSD

IterPSD outperforms Naive-SemiICL across five classification tasks, as shown in Table 1. We evaluate both methods using Verbalized Confidence and Self-Consistency. Notably, IterPSD achieves significant gains on BANKING, CLINC, CLINC(D), and FewEvent (over 3.0% performance gain), but not on FP. Similar to Naive-SemiICL, we observe a scaling law with respect to the number of pseudo-demonstrations used in IterPSD. Clear scaling trends are observed in four out of five tasks, as shown in Figure 3. On these tasks, IterPSD attains peak performance with 500 to 1,000 pseudo-demonstrations. The lack of scaling on FP may be attributed to the relative ease of the dataset, as Naive-SemiICL already achieved 98% accuracy on this task.

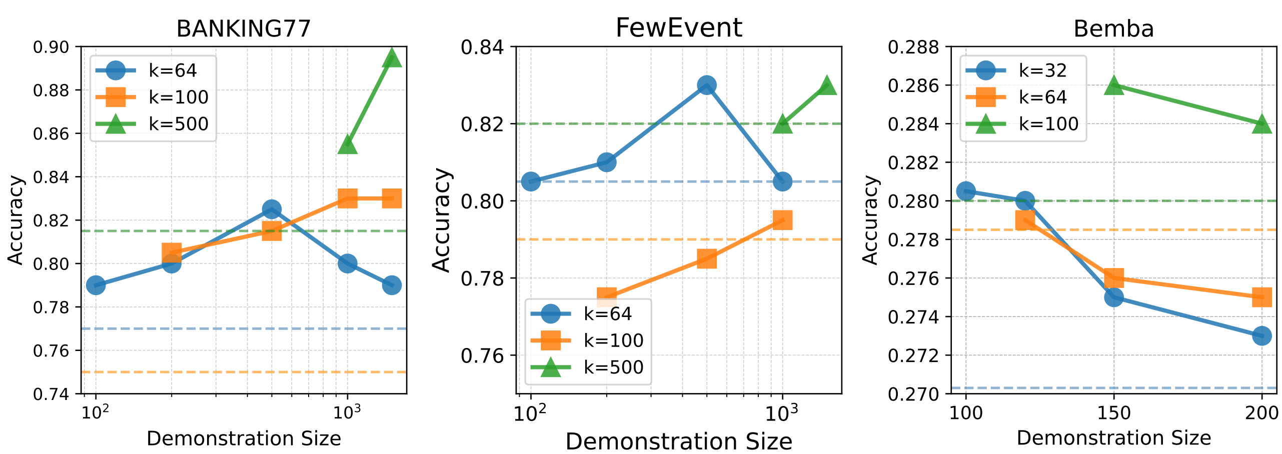

Figure 5: We compare Naive-SemiICL accuracy across different ground truth demonstration sizes, with baseline performances indicated by dashed lines. On FewEvent, the maximum number of pseudo-demonstrations is capped at 1000 due to the limited availability of pseudo-demonstrations after filtering.

We also benchmark IterPSD on translation tasks, but the improvement over Naive-SemiICL is not consistent. We attribute this to the fact that each iteration of IterPSD needs to accumulate at least 100 demonstrations to avoid bias from sampling noise. However, Semi-Supervised ICL typically degrades after approximately 200 demonstrations, resulting in IterPSD terminating after 2 to 3 iterations.

Method	GPQA	Math	Logical7	Shapes	Date
Naive-SemiICL	42.42	**40.78**	**90.00**	**78.00**	**79.00**
MoT	**44.44**	25.86	88.00	64.00	58.00
Reinforced ICL	54.54	42.63	93.00	78.00	89.00

Table 2: Comparison of Naive-SemiICL (Naive) and MoT on reasoning datasets using GPT-4o-mini.

E.3. Comparing Naive-SemiICL & MoT

On reasoning datasets, Naive-SemiICL outperforms MoT on all tasks except GPQA, as shown in Table 2. Surprisingly, the performance gap between the two methods is substantial on LiveBench Math, Shapes, and Date. We attribute this to two key differences between Naive-SemiICL and MoT: (1) MoT uses Entropy to filter low-quality demonstrations, which we show to be less reliable than Verbalized Confidence (see Table 3); and (2) in preliminary experiments, we found similarity-based retrieval to be less effective than diverse sampling. Naive-SemiICL samples diversely from a large pool of pseudo-demonstrations, which MoT is unable to do due to its requirement to query the LLM for each demonstration retrieval.

		GPT-4o-mini				GPT-4o			
Task Type	**Task**	**Verbalized**	**Self-Consistency**	**Entropy**	**Back-Translation**	**Verbalized**	**Self-Consistency**	**Entropy**	**Back-Translation**
Classification	BANKING	$\mathbf{75.33 \pm 0.20}$	75.16 ± 0.20	-	-	72.17 ± 0.20	$\mathbf{72.30 \pm 0.20}$	-	-
	CLINC	89.16 ± 0.80	$\mathbf{91.17 \pm 0.40}$	-	-	95.50 ± 0.70	$\mathbf{95.80 \pm 0.90}$	-	-
	CLINCD	66.33 ± 0.50	$\mathbf{69.17 \pm 0.20}$	-	-	$\mathbf{79.33 \pm 0.20}$	77.80 ± 0.20	-	-
	FewEvent	69.33 ± 0.50	$\mathbf{73.33 \pm 0.20}$	-	-	76.17 ± 0.50	$\mathbf{77.17 \pm 0.20}$	-	-
	FP	97.50 ± 0.50	$\mathbf{97.83 \pm 0.20}$	-	-	96.50 ± 0	$\mathbf{97.83 \pm 0.20}$	-	-
	AVG	79.53	**81.33**	-	-	83.93	**84.18**	-	-
Translation	Bemba	$\mathbf{27.93 \pm 0.10}$	-	26.66 ± 0.20	27.42 ± 0.30	$\mathbf{29.16 \pm 0.20}$	-	27.65 ± 0.20	28.34 ± 0.20
	Fijian	$\mathbf{36.70 \pm 0.20}$	-	35.96 ± 0.10	36.14 ± 0.10	$\mathbf{42.67 \pm 0.40}$	-	41.42 ± 0.30	41.98 ± 0.40
	Faroese	$\mathbf{43.97 \pm 0.20}$	-	42.32 ± 0.20	43.95 ± 0.20	$\mathbf{49.69 \pm 0.40}$	-	48.01 ± 0.40	48.93 ± 0.30
	Venetian	$\mathbf{44.41 \pm 0.20}$	-	43.84 ± 0.10	43.26 ± 0.20	$\mathbf{45.05 \pm 0.30}$	-	44.53 ± 0.50	44.67 ± 0.40
	Tuvan	$\mathbf{19.61 \pm 0.30}$	-	19.53 ± 0.10	19.02 ± 0.20	$\mathbf{23.75 \pm 0.30}$	-	23.01 ± 0.30	22.57 ± 0.40
	Sardinian	$\mathbf{41.27 \pm 0.20}$	-	40.53 ± 0.10	40.63 ± 0.20	$\mathbf{47.94 \pm 0.20}$	-	46.82 ± 0.10	47.85 ± 0.30
	AVG	**35.65**	-	34.81	35.07	**39.71**	-	38.57	39.06
Reasoning	GPQA	40.40 ± 0.50	$\mathbf{42.42 \pm 0.50}$	41.41 ± 0.50	-	$\mathbf{52.52 \pm 0.50}$	47.47 ± 0.50	52.52 ± 0.50	-
	LB Math	$\mathbf{40.78 \pm 0.30}$	35.52 ± 0.50	35.48 ± 0.30	-	36.33 ± 0.80	$\mathbf{39.78 \pm 0.30}$	30.10 ± 0.30	-
	logical7	$\mathbf{90.00} \pm 0.50$	84.00 ± 0	86.00 ± 0.50	-	98.00 ± 0.50	$\mathbf{100.00 \pm 0.50}$	100.00 ± 0.50	-
	Geometric	70.00 ± 0	66.00 ± 0	$\mathbf{78.00 \pm 0.50}$	-	61.00 ± 0	67.00 ± 0	$\mathbf{70.00 \pm 0.50}$	-
	Date	$\mathbf{42.00 \pm 0.80}$	32.00 ± 0	35.00 ± 0	-	$\mathbf{68.00 \pm 0.80}$	65.00 ± 0	67.00 ± 0.50	-
	AVG	**56.64**	51.99	55.18	-	63.17	63.85	**63.92**	-

Table 3: Comparison of GPT-4o-mini and GPT-4o performance using different confidence scores. Each task is evaluated using different inference strategies: Verbalized, Self-Consistency, Entropy, and Back-Translation (where applicable). Reported values on represent average accuracy and ChrF++ with standard deviations.

F. Effects of Different Confidence Methods

In this section, we examine the performance Naive-SemiICL paired with different confidence methods, which we compile as Table 3. We observe that classification and translation tasks each have a dominant confidence measure. For classification tasks, Self-Consistency emerges as the most effective confidence method. It surpasses the Verbalized Confidence method on 4 out of 5 datasets across both models. Verbalized Confidence is the leading measure for translation tasks, consistently achieving the highest performance across all languages. For reasoning tasks, no single method clearly dominates. Under GPT-4o-mini, Verbalized Confidence yields the best average performance, while under GPT-4o, Entropy slightly outperforms Self-Consistency, securing the top position by a narrow margin.

Overall, Self-Consistency improves classification and reasoning tasks, but its effect varies across translation tasks and is not applicable to all tasks. Entropy is sometimes useful in reasoning tasks, but fall short on translation tasks. Verbalized inference remains a strong and economical baseline across all tasks but is generally outperformed by Self-Consistency on classification tasks.

F.1. Analyzing Performance Decline of Naive-SemiICL

We hypothesize that Naive-SemiICL's decline in performance beyond a certain demonstration size stems from the accumulation of errors in pseudo-demonstrations. To isolate the negative impact of long contexts on the LLMs, we examine the scaling behavior when all demonstrations are ground-truth data. Figure 6 shows that both GPT-4o-mini and GPT-4o continue to improve as the number of demonstrations increases, even beyond the optimal demonstration size for Naive-SemiICL in the 16-shot setting. This suggests that the performance degradation is not caused by long context length, but rather by the accumulated errors in pseudo-demonstrations. This finding motivates the design of IterPSD, which addresses error accumulation in pseudo-annotations through curriculum learning and iterative refinement.

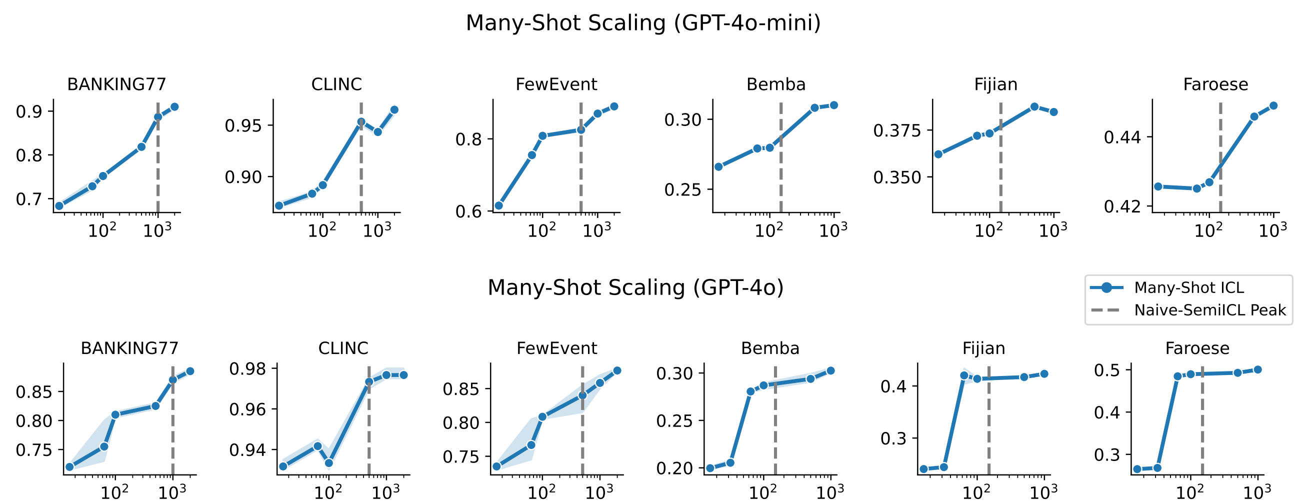

Figure 6: Many-shot scaling performance of GPT-4o-mini (top) and GPT-4o (bottom) across six selected datasets. The x-axis represents the number of shots (log scale), and the y-axis represents performance. The solid blue lines indicate many-shot in-context learning (ICL), while the dashed vertical lines mark the peak performance of Naive-SemiICL. Both models scale beyond the peak the performance of pseudo-demonstration approach.

G. Related Work

Self-Generated Demonstrations. Large Language Models (LLMs) exhibit remarkable zero-shot capabilities, allowing them to perform tasks without task-specific fine-tuning or prior examples. Their zero-shot predictions have proven to be effective sources of demonstration for in-context learning (Kojima et al., 2022; Zou et al., 2025a).

Auto-CoT (Zhang et al., 2023) prompts the LLM with self-generated rationales on diversely sampled inputs. Rationales consisting of more than five reasoning steps are excluded from the demonstration to maintain the simplicity and accuracy of the demonstration. Such task-specific heuristic does not generalize to most recently published datasets such as LiveBench Math, as most of the generated rationales contain more than five steps. (Li & Qiu, 2023) builds on top of Auto-CoT with extra an extra step of semantic filtering. At each example during inference, the LLM is prompted to choose the demonstration for itself after retrieving the semantically relevant demonstrations through an embedding model. Like Auto-CoT, Reinforced ICL (Agarwal et al., 2024) generates rationales for reasoning problems and filters out those leading to incorrect answers. While this method requires ground truths, our filtering method do so with self-generated confidence score.

PICLe (Mamooler et al., 2024) generates new demonstrations by annotating unlabeled examples and filtering out those with incorrect named entity types through self-verification prompting. Similarly, SAIL (Li et al., 2024a) employs an annotation strategy for the bilingual lexical induction task, discarding predictions that fail to translate back to the original input. Both methods rely on task-specific filtering and require additional LLM queries for self-verification or back-translation. In contrast, our Verbalized Confidence approach is task-agnostic and requires only a single prompt for pseudo-labeling, significantly reducing inference overhead. Z-ICL (Li et al., 2024b) leverages the zero-shot generative capability of large language models to synthesize demonstrations for subsequent in-context learning inference. In contrast, our approach assumes access to abundant unlabeled data and a small set of ground-truth labels, using the LLM only for annotation rather than for input generation.

Many-Shot ICL. (Agarwal et al., 2024) observed a significant performance increase in a variety of generative and discriminative tasks, as well as a scaling law between the number of examples in the demonstration and ICL performance. Our method hinges on this ability as our proposed method, Naive-SemiICL, fits at least 64 examples in the prompt. We report a similar scaling law for Semi-Supervised ICL in this work.

Traditional Semi-Supervised Learning. Semi-supervised learning seeks to reduce reliance on labeled data by leveraging abundant unlabeled data to enhance model performance (Lee et al., 2013; Sohn et al., 2020; Zou et al., 2025b). Self-training (McLachlan, 1975; Xie et al., 2020) iteratively refines the model by using its own predictions on unlabeled data for training. Pseudo-labeling (Lee et al., 2013; Sohn et al., 2020; Zou et al., 2023a;b) employs confidence-based filtering, retaining only high-confidence pseudo-labels to reduce error propagation and confirmation bias. JointMatch (Zou & Caragea, 2023) further alleviates error accumulation by using two independently initialized networks that teach each other through cross-labeling. Our work is the first to integrate confidence filtering and leverage both labeled and pseudo-labeled data in an in-context learning framework.

H. Impact Statement

This work investigates scalable semi-supervised approaches for in-context learning using self-generated pseudo-demonstrations. Our methods, Naive-SemiICL and IterPSD, reduce reliance on labeled data and enable effective use of large language models in low-resource settings. These techniques have the potential to broaden access to high-quality language technologies across domains and languages with limited annotation resources.

However, the use of self-generated data raises concerns about the propagation of biases or errors inherent in the language model. Careful filtering and evaluation are necessary to mitigate unintended effects, particularly in sensitive applications. We encourage future research on aligning pseudo-demonstrations with human intent and fairness objectives.

Table 4: The prompt template we use for classification, translation, and reasoning tasks, respectively.

Types	Prompts
Classification	You are a helpful assistant who is capable of performing a classification task (mapping an Input to a Label) with the following possible labels: {A LIST OF POSSIBLE LABELS} --- Here are zero or more Input and Label pairs sampled from the classification task. {DEMONSTRATIONS} --- Now, Label the following Input among the following Input: {INPUT}
Translation	You are an expert translator. I am going to give you zero or more example pairs of text snippets where the first is in the source language and the second is a translation of the first snippet into the target language. The sentences will be written in the following format: ¡source language¿: ¡first sentence¿ ¡target language¿: ¡translated first sentence¿ --- {DEMONSTRATIONS} --- Now, Translate the following $source text into $target. Also give the Confidence of your given Answer in the following format: **Confidence**: ¡a confidence score between 0 and 1¿: English: {INPUT SENTENCE} {TARGET LANGUAGE}:
Reasoning	First, I am going to give you a series of Questions that are like the one you will be solving. --- {DEMONSTRATIONS} --- Now, Answer the following Question. Think step by step. Question: {QUESTION} Also give the Confidence of your given Answer in the following format: **Confidence**: ¡a confidence score between 0 and 1¿

Algorithm 2 Naive-SemiICL.

1: **Input:** prompt ρ, ground-truth demonstrations $\mathcal{E}_g$, unlabeled data $\mathcal{X}_u$;
2: Initialize $\mathcal{D}_{\mathrm{PSD}} = \emptyset$;
3: **for** $x \in \mathcal{X}_u$ **do**
4: $\hat{y}, \hat{c} = \mathrm{LM}(\rho_{\mathcal{T}}, \mathcal{E}_l, x)$;
5: $\mathcal{D}_{\mathrm{PSD}} = \mathcal{D}_{\mathrm{PSD}} \cup \{(x, \hat{y}, \hat{c})\}$;
6: **end for**
7: **Return** $\mathcal{D}_{\mathrm{PSD}}$;

Algorithm 3 ϵ-Random Sampler

1: **Input:** annotated demonstration $\mathcal{D}_l$, un-annotated demonstration $\overline{\mathcal{D}}_l$, chunk size K, random ratio ϵ, prompt ρ, embedder ϕ.
2: Initialize $S = \emptyset$;
3: $K_{\mathrm{random}} = \epsilon K, K_{\mathrm{sim}} = (1-\epsilon)K$;
4: Compute $d_{ij} = \mathrm{sim}_{\cos}(\phi(x_i), \phi(x_j))$ for all $x_i \in \mathcal{D}_l, x_j \in \overline{\mathcal{D}}_l$;
5: Compute $d_j = \min_i d_{ij}$ for all $x_j \in \overline{\mathcal{D}}_l$;
{*Compute distance to the nearest annotated example.*}
6: $S_{\mathrm{sim}} = \{x_j | d_j \in \text{Smallest}_{K_{\mathrm{sim}}}\{d_j\}\}$;
{*select the K_{sim} examples with the smallest distance to its nearest annotated demonstrations*}
7: Compute S_{random}, a random sample of size K_{random} from $\overline{\mathcal{D}}_l - S_{\mathrm{sim}}$);
8: $S = S_{\mathrm{sim}} \cup S_{\mathrm{random}}$;
9: **Return** S;