Many-Shot Scaling of In-Context Learning with Self-Generated Demonstrations

Anonymous ACL submission

Abstract

The high cost of obtaining high-quality annotated data for in-context learning (ICL) has motivated the development of methods that use *self-generated annotations* in place of ground truth labels. While these approaches have shown promising results in few-shot settings, they generally do not scale to many-shot scenarios. In this work, we study ICL with self-generated examples using a framework analogous to traditional semi-supervised learning, consisting of annotation generation, demonstration selection, and in-context inference. Within this framework, we propose a simple baseline that outperforms ground truth ICL under zero-shot, few-shot, and many-shot settings. Notably, we observe consistent *scaling* behaviors with respect to the number of self-annotated demonstrations. To further extract performance from this many-shot capability, we introduce IterPSD, an iterative self-annotation approach that integrates iterative refinement and curriculum pseudo-labeling techniques from semi-supervised learning, yielding up to 6.8% additional gains on classification tasks. Motivated by our baseline and IterPSD results, we demonstrate that semi-supervised ICL offers a promising avenue for future ICL research. Code is available at `https://anonymous.4open.science/r/semi-supervised-icl-FB8B`.

1 Introduction

In-context learning (ICL) has emerged as a powerful paradigm in natural language processing, enabling large language models (LLMs) 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 and 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,

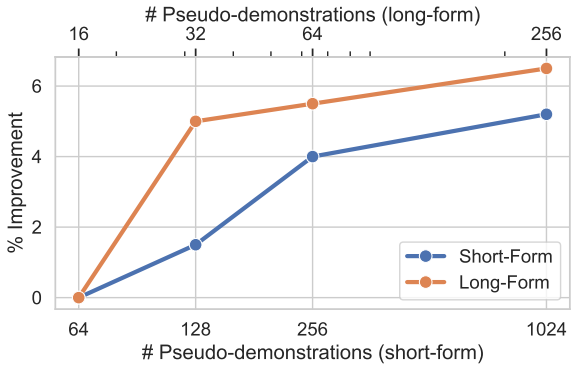

Figure 1: Average performance improvement in long-form (translation, reasoning) and short-form tasks (classification) with respect to number of pseudo-demonstrations used. Performance improvement on self-generated demonstrations scales beyond few-shot settings. Results obtained from GPT-4o.

(Agarwal et al., 2024) observed many-shot scaling with ground truth labels, 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 2): 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 under zero-shot, few-shot, and many-shot ground truth budgets. Additionally, we observe a consistent scaling behavior between ICL performance and number of self-annotated demonstrations across short-form and long-form generation tasks (Figure 1).

With potentially thousands of self-annotated examples in the prompt, each prompt 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 self-annotation approach that progressively refines pseudo-demonstration quality by incorporating self-generated annotations into the existing prompt. IterPSD further improves semi-supervised ICL performance on five classification tasks, achieving gains of up to 6.8%.

2 Method

In this section, we establish the framework of semi-supervised ICL, which consists of three phases: ① pseudo-demonstration generation, ② demonstration selection, and ③ semi-supervised inference. We then propose a simple baseline for semi-supervised ICL, Naive-SemiICL, which generates pseudo-demonstrations in a single iteration and filters out examples with low confidence scores. Building on Naive-SemiICL, we introduce an iterative method, IterPSD, that progressively improves the prompt by incorporating self-generated annotations during the demonstration generation process.

2.1 Semi-Supervised ICL

Confidence-Aware In-Context Learning extends traditional ICL by outputting an additional confidence score c for each input:

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

Like traditional ICL, the LLM is prompted with a prompt ρ associated with the task, a set of demonstrations $\mathcal{E}$, and an input x. The confidence score c provides a measure of certainty for its output y. We define the prediction y broadly in this setting. y could be a single predicted label in a classification task. Or it could could contain both a predicted answer and a rationale that leads to the answer (Wei et al., 2022)

$$y := (\hat{y}, r). \tag{2}$$

Pseudo-Demonstration Annotation. During step ①, Semi-supervised ICL annotates large set of *unannotated data* $\mathcal{X}_u = \{x_i\}^{k_u}$, using a small set of *ground truth data* $\mathcal{E}_g = \{(x_i, y_i)\}^{k_g}$ (or none) as demonstrations. We denote the resulting set of annotations as

$$\mathcal{D}_{\text{PSD}} = \{(x, y, c) | x \in \mathcal{X}_u\}, \tag{3}$$

where y and c are generated from Equation 1. We then sample pseudo-demonstrations from annotations whose confidence surpasses some threshold $c \geq \lambda$, which is assumed to be uniformly random unless otherwise stated.

$$\mathcal{E}_u = \text{Sampler}(\mathcal{D}_{\text{PSD}}, \lambda) \tag{4}$$

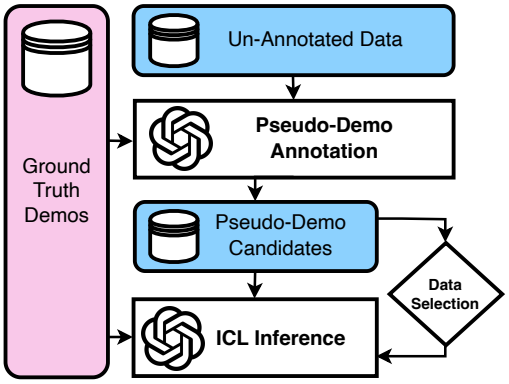

Figure 2: Semi-supervised ICL Framework.

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

$$y = \text{LLM}(\rho, \mathcal{E}_u \cup \mathcal{E}_g, x) \tag{5}$$

Unlike ICL, where all used demonstrations are constructed from ground truth data, semi-supervised ICL uses both ground truth and self-annotated data, analogous to traditional semi-supervised learning. Departing from previous work on ICL that explores LLM's ability to self-annotated demonstrations (Li and Qiu, 2023; Zhang et al., 2023), semi-supervised ICL emphasizes how different budgets of ground truth data impacts model performances.

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 Naive-SemiICL in Algorithm 1.

2.3 Iterative Pseudo-Demonstration Generation

Encouraged by the success of Naive-SemiICL, we explore whether pseudo-demonstrations can enhance the accuracy of subsequent pseudo-demonstration generation. We propose **IterPSD** (Algorithm 2), an iterative method for generating pseudo-demonstrations that:

1. recursively adds newly generated pseudo-demonstrations to its prompt until reaching the maximum number of allowed demonstrations (Line 13), and

2. re-samples the most confident pseudo-demonstrations according to a confidence threshold λ from all previously annotated instances once the demonstration size reaches its limit (Line 7).

Algorithm 1 Naive-SemiICL.

Input: prompt ρ, ground-truth demonstrations $\mathcal{E}_g$, unlabeled data $\mathcal{X}_u$;
Initialize $\mathcal{D}_{\text{PSD}} = \emptyset$;
for $x \in \mathcal{X}_u$ **do**
  $\hat{y}, \hat{c} = \text{LLM}(\rho_{\mathcal{T}}, \mathcal{E}_l, x)$;
  $\mathcal{D}_{\text{PSD}} = \mathcal{D}_{\text{PSD}} \cup \{(x, \hat{y}, \hat{c})\}$;
end for
Return $\mathcal{D}_{\text{PSD}}$;

Algorithm 2 IterPSD

1: **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 pseudo-demonstrations $\text{Sampler}_{\text{PSD}}$;
2: Initialize $\mathcal{D}_{\text{PSD}} = \emptyset$; {*Set of all the annotated pseudo-demonstrations.*}
3: Initialize $\overline{\mathcal{D}}_{\text{PSD}} = \mathcal{X}_u$; {*Set of data yet to be annotated.*}
4: Initialize $\mathcal{E} = \mathcal{E}_g$; {*Demonstration for generating pseudo-demonstrations.*}
5: **while** $\overline{\mathcal{D}}_{\text{PSD}} \neq \emptyset$ **do**
6: **if** $|\mathcal{E}| > \kappa$ **then**
7: $\mathcal{E} =$ top-κ confident examples in $\mathcal{D}_{\text{PSD}}$; {*Cap the demonstration at a maximum size*}
8: **end if**
9: $S_u = \text{Sampler}_{\epsilon}(\mathcal{D}_{\text{PSD}}, \overline{\mathcal{D}}_{\text{PSD}}, K, \epsilon)$; {*Retrieves a sample of size K using ϵ-Random Sampler*}
10: $S_{\text{PSD}} = \text{Naive-SemiICL}(S_u, \rho, \mathcal{E})$; {*Label S_u with one iteration of Naive-SemiICL.*}
11: $S^{\lambda}_{\text{PSD}} = \text{Filter}(S_{\text{PSD}}, \lambda)$;
12: $\mathcal{E} = \mathcal{E} \cup S^{\lambda}_{\text{PSD}}$;
13: $\mathcal{D}_{\text{PSD}} = \mathcal{D}_{\text{PSD}} \cup S_{\text{PSD}}$;
14: $\overline{\mathcal{D}}_{\text{PSD}} = \overline{\mathcal{D}}_{\text{PSD}} - S_{\text{PSD}}$;
15: **end while**
16: **Return** $\mathcal{D}_{\text{PSD}}$;

In each iteration, IterPSD samples and annotates K unlabeled examples before applying a filtering step. The generated pseudo-demonstrations are recursively accumulated and fed back into the LLM to generate additional pseudo-demonstrations (Line 10). To mitigate performance degradation as more erroneous pseudo-demonstrations are incorporated, we impose an upper limit κ on the number of self-fed pseudo-demonstrations. Once this limit is reached, we resample the κ most confident pseudo-demonstrations from all existing pseudo-demonstrations, ensuring that only the highest quality examples are utilized for subsequent self-annotation.

Curriculum Learning. To further improve the accuracy of self-annotation, we sample both similar and diverse examples from the unannotated pool for each iteration of self-annotation. At each iteration, a proportion $(1 - \epsilon)$ of examples is chosen to be the nearest previously annotated instances based on some similarity measure. This ensures that each selected example is similar to an existing annotation. The remaining examples are chosen from diverse clusters, similar to the strategy in (Zhang et al., 2023). By prioritizing similar examples, we create a structured progression from easy to hard cases, following a curriculum learning paradigm (Soviany et al., 2021). This schedule allows the model to first adapt to examples that closely resemble annotated data before being exposed to more challenging or atypical instances. As a result, the annotation process achieves lower error rates compared to uniform or random sampling strategies. We dub this family of samplers ϵ-Random Samplers. We present the detailed algorithm in Appendix D.

Mitigating Confirmation Bias. To maintain annotation quality, we find that at least half of the data ($\epsilon \geq 0.5$) should be sampled diversely. When $\epsilon = 0$, selections are exclusively based on similarity to previously annotated examples, and the pseudo-demonstrations become homogeneous, leading to bias in ICL predictions. This phenomenon closely parallels confirmation bias in semi-supervised learning (Arazo et al., 2019; Zou and Caragea, 2023).

3 Experimental Setup

Tasks and Datasets. Our evaluation consists of 16 datasets spanning 9 tasks and 3 task types:

- **Classification.** We include BANKING77 (Casanueva et al., 2020), CLINC, CLINC(D) (Larson et al., 2019), FewEvent (Deng et al., 2020), and FP (Malo et al., 2013).
- **Translation.** We evaluate Naive-SemiICL's ability to translate English into low-resource languages using 6 datasets from FLORES200: Bemba, Fijian, Faroese, Tuvan, Venetian, and Sardinian (Costa-Jussà et al., 2022).
- **Reasoning.** We include 5 benchmarks spanning scientific, mathematical, and logical reasoning: GPQA (Rein et al., 2024), LiveBench Math (White et al., 2025), and three tasks from BigBenchHard (Suzgun et al., 2022): Logical7, Geometric Shapes, and Date.

We describe these datasets in detail in Appendix B.3 and explain how we split the training and testing data in Appendix B.1.

Evaluation Metrics. For all classification and reasoning tasks, we report **accuracy** as the performance metric. We evaluate the equivalence of LaTeX-style mathematical outputs on LiveBench Math using the parser described in (Gao et al., 2024). For translation tasks, we report the **ChrF++** score (Popović, 2015) using its default configuration, as implemented in TorchMetrics (Detlefsen et al., 2022), following (Agarwal et al., 2024). We report the mean and standard error over three trials

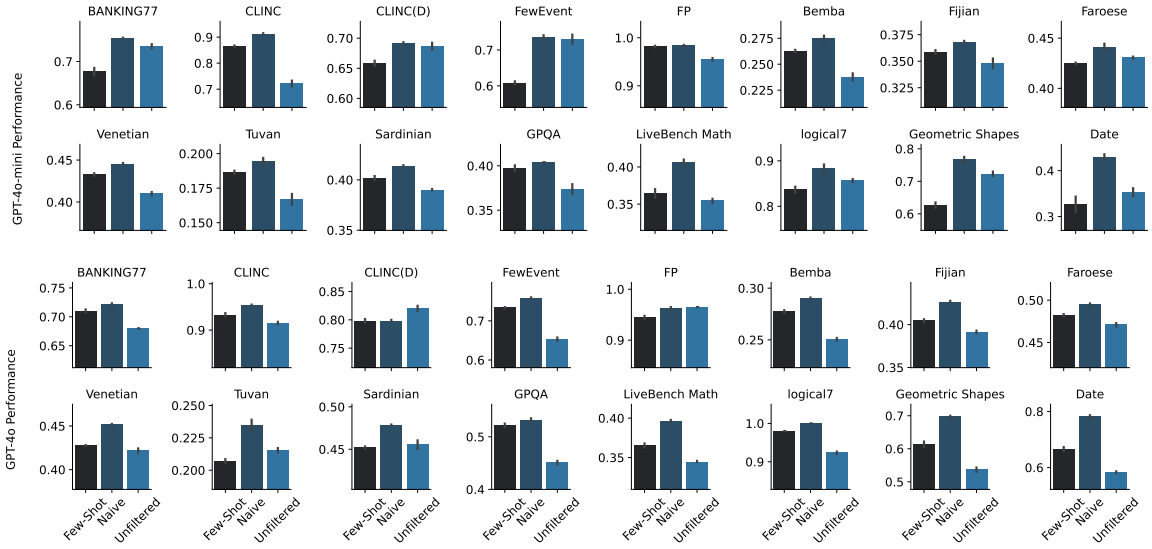

Figure 3: Comparison of GPT-4o-mini (top) and GPT-4o (bottom) performance across multiple datasets using three different methods: Few-Shot, Naive-SemiICL (Naive), and Naive-SemiICL without filtering (Unfiltered).

	Task	Zero-shot	Naive	Improv.
Classification	Banking	61.50	**78.00**	26.80%
	FewEvent	56.00	**65.00**	16.07%
	CLINC	83.50	**88.50**	5.99%
	CLINCD	59.50	**61.00**	2.52%
	FP	91.00	**94.50**	3.85%
Translation	Bemba	0.2437	**0.2591**	12.37%
	Fijian	33.63	**35.16**	4.55%
	Faroese	42.45	**42.90**	1.06%
	Venetian	42.03	**42.82**	1.88%
	Tuvan	16.17	**18.40**	13.79%
	Sardinian	37.06	**38.46**	3.78%
Reasoning	GPQA	36.36	**38.38**	5.55%
	Math	35.48	**36.58**	3.10%
	Logical7	65.00	**72.00**	10.77%
	Shapes	56.00	**60.00**	7.14%
	Date	40.00	**65.00**	62.5%

Table 1: Performance comparison of Zero-shot ICL and Naive-SemiICL. All experiments are done on GPT-4o-mini with Verbalized Confidence. For Naive-SemiICL, we report the result from the best performing number of pseudo-demonstrations.

for baseline results(Section 4.1). The remaining results are based on a single trial.

Demonstration Budget. Based on the number of ground truth demonstrations used, we divide our experiments into zero-shot, few-shot, and many-shot settings, where the number of available ground truth demonstrations varies. In the zero-shot setting, no ground truth demonstrations are used. In the few-shot setting, $k_g = 16$ ground truth demonstrations are allowed. In many-shot setting, $k_g \in \{32, 64, 100, 500\}$ ground truth data are available. We experiment with different pseudo-demonstration sizes: $k_u \in \{32, 64, 128, 512, 1024, 2048\}$ for classification tasks, $k_u \in \{32, 64, 128, 256\}$ for translation and reasoning tasks.

Baselines. We use the following methods as baselines.

- **k-Shot ICL.** The LLM is prompted with k ground truth annotated examples.
- **Unfiltered SemiICL.** To highlight the importance of confidence-based data selection, we include an unfiltered variant of Naive-SemiICL, which samples pseudo-annotations without applying the filtering step.
- **MoT.** (Li and Qiu, 2023) We include MoT as a domain-specific baseline for reasoning tasks. Unlike MoT, Naive-SemiICL uses a simple one-step filtering mechanism for demonstration selection, whereas MoT requires querying the LLM for each example. We detail how we configure MoTin Appendix B.4.
- **Reinforced ICL.** (Agarwal et al., 2024) demonstrates that prompting the LLM with self-generated reasoning chains filtered by ground truth answers can significantly improve ICL performance. This method serves as an upper bound on semi-supervised ICL performance on the reasoning tasks when the filtering mechanism is assumed to be perfect.

During our preliminary experiments, we found Auto-CoT (Zhang et al., 2023) to be uncompetitive on our reasoning datasets, as it relies on simple heuristics for data selection that are no longer effective. Since MoT includes all of Auto-CoT's steps except its entropy- and semantic-based filters, we opted not to report Auto-CoT performances.

	Bemba	BANKING77	FewEvent
$k_g = 32$	27.08/**28.04**	-	-
$k_g = 64$	27.85/**27.92**	75.00/**83.00**	79.00/**79.50**
$k_g = 100$	28.01/**28.60**	77.00/**82.50**	80.50/**83.00**
$k_g = 500$	-	81.50/**89.50**	82.00/**83.50**

Table 2: Performance of Naive-SemiICL on three datasets, compared to ICL performance with a many-shot ground truth budget. Each cell follows the format (baseline / Naive-SemiICL).

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 3: Comparison of Naive-SemiICL (Naive) and MoT on reasoning datasets using GPT-4o-mini. Experiment conducted with $k_g = 3$. Reported performance reflects the optimal number of pseudo-demonstrations for each method.

Confidence Scores. We primarily evaluate three confidence metrics: Verbalized Confidence, which prompts the LLM to generate the confidence score, Entropy, and Self-Consistency. Self-Consistency measures the confidence as the frequency of the most frequent answer, and entropy is defined as

$$c_{\text{Ent}} = -\frac{1}{L}\sum_{i=s}^{L} \log P(w_i \mid w_{<i}). \quad (6)$$

We detail how we prompt the LLM for Verbalized Confidence in Appendix A, and how we apply the confidence measures in Appendix A and B.5.

Hyperparameters. In our experiments, we performed hyperparameter tuning over the confidence threshold λ using values at $80, 85, 90, 95$-th percentiles on a subset of tasks. We found that setting the confidence threshold at 90% is generally performant across our benchmark. For the remaining tasks, we fix this threshold without additional tuning.

For IterPSD, we experimented with random sample proportion $\epsilon \in \{0.5, 0.8, 0.9, 0.95\}$ and found $\epsilon = 0.8$ to be most performant on most tasks, except on FP where $\epsilon = 0.5$ is optimal. Similarly, we found $K = 500$ to be performant on all tasks except FP, where $K = 100$ yielded the best results. During our preliminary experimentation, we found the maximum pseudo-demonstration size κ should be chosen slightly smaller than the optimal pseudo-demonstration size for Naive-SemiICL, which turns out to be 1000 for most classification tasks. More details on hyperparameters can be found in Appendix B.7.

Models. We experiment with GPT-4o-mini and GPT-4o, checkpointed on 2024-07-18 and 2024-11-20, respectively. For the ϵ-random sampler, we compute similarity using vector embeddings generated by OpenAI's *text-embedding-3-large*. We discuss the computational cost associated with our experiments in Appendix B.2.

4 Empirical Analyses

4.1 Naive-SemiICL Consistently Beats Baselines

We first compare the performance of Naive-SemiICL with Verbalized Confidence to the few-shot baseline in Figure 3. In these settings, we use $k_g = 16$ ground truth demonstrations. For Naive-SemiICL, we report performance using the optimal number of pseudo-demonstrations k_u for each task. The best-performing k_u values are shown in Appendix B.4. We can see in Figure 3 that Naive-SemiICL outperforms few-shot ICL on all tasks except CLINC(D), where it matches the baseline. Unfiltered SemiICL fails to match baseline performance in 20 out of 32 settings, highlighting the importance of the filtering step. We provide additional experimental results of Naive-SemiICL using different confidence scores in Appendix C.

We demonstrate the effectiveness of Naive-SemiICL in low-resource settings using a zero-shot experimental setup, with results shown in Table 1. The performance gap between Naive-SemiICL and the zero-shot baseline depends solely on the quality of the filtering mechanism. As shown in Table 1, Naive-SemiICL outperforms the zero-shot baseline on all tasks in the benchmark, attaining an average improvement of 11.36% under GPT-4o-mini. This exceeds the average improvement of 9.94% in the 16-shot setting (Figure 3), suggesting that Naive-SemiICL is more effective in resource-constrained conditions.

Additionally, we found Naive-SemiICL to be effective in high-resource settings. Table 2 compares the performance of Naive-SemiICL and ICL 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, while $k_g = 500$ only yields a 4.73% improvement across the three tasks. This trend supports the earlier conclusion that semi-supervised ICL is more effective in low-resource settings. Taken together, these results highlight the consistent effectiveness of Naive-SemiICL across varying budgets of ground truth data.

We compare self-annotation methods that are specifically tailored for reasoning tasks in Table 3. We adopt the few-shot settings. Naive-SemiICL outperforms MoT on all tasks except GPQA. 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

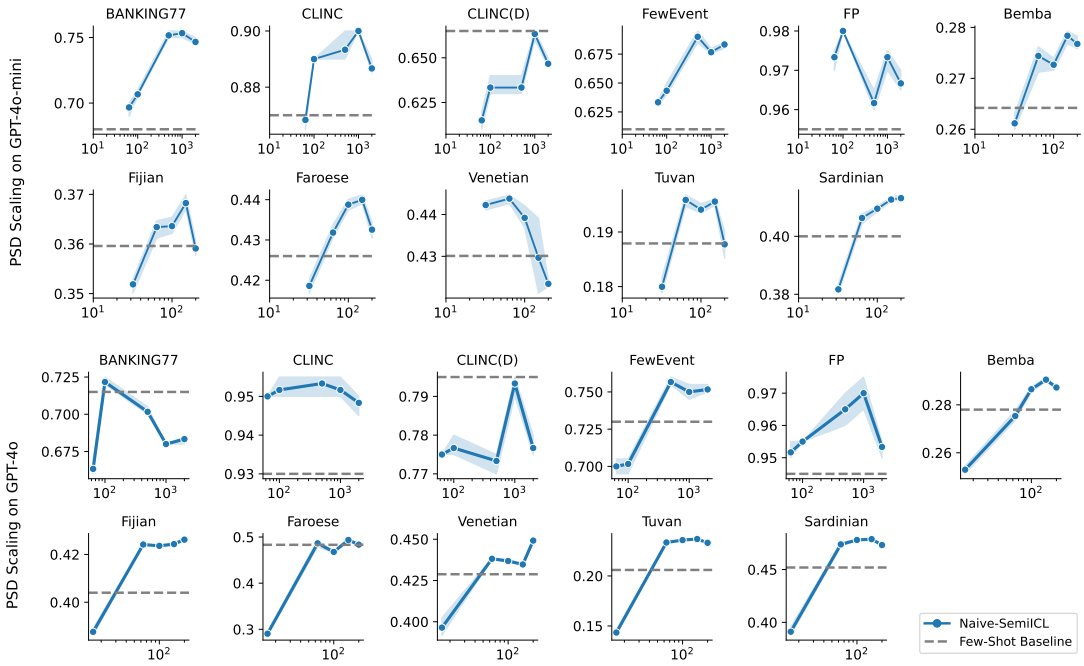

Figure 4: PSD (pseudo-demonstration) scaling trend of Naive-SemiICL on classification and translation tasks with GPT-4o and GPT-4o-mini. The dashed gray line represents the few-shot baseline.

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 4: 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.

Confidence in Appendix C; and (2) in preliminary experiments, we found similarity-based retrieval to be less effective than diverse sampling for Naive-SemiICL. Naive-SemiICL samples diversely from a large pool of pseudo-demonstrations, whereas MoT queries a LLM to select the most relevant demonstrations, missing out the benefit brought by diversity. Additionally, the query-based demonstration selection mechanism is only economically viable when a few pseudo-demonstrations are used. The practice would incur prohibitive costs when potentially incorporating thousands of pseudo-demonstrations is beneficial.

4.2 Semi-Supervised ICL Scaling

We observe consistent scaling trends for semi-supervised ICL, similar to the one reported in many-shot ICL (Agarwal et al., 2024). We illustrate this trend in Figure 4 using a ground truth demonstration budget of 16. Across all tasks, Naive-SemiICL performance improves with larger demonstration sizes, although the point of peak performance varies. Both GPT-4o and GPT-4o-mini scale effectively across most tasks, typically peaking between 500 and 1,000 examples for classification tasks and between 100 and 200 examples for translation tasks. A similar scaling trend is observed with no ground truth demonstrations in Figure 6 across both classification and translation tasks. We observe that the scaling trend peaks at higher numbers of pseudo-demonstrations under the zero-shot setting, typically around 2000 pseudo-demonstrations compared to 500 in the 16-shot setting. This suggests semi-supervised ICL is more effective when the ground truth budget is stringent. We also observe scaling trends under high ground-truth budgets. Figure 8 demonstrates the scaling effect when $k = \{64, 100, 500\}$ ground truth demonstrations are used. In all, we demonstrate that self-annotated demonstrations can provide extra training signals even when ground truth data is abundant.

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 (Liu et al., 2023; Anil et al., 2022), we examine the scaling behavior when all demonstrations are ground truth data. Figure 7 shows that both GPT-4o-mini and GPT-4o continue to improve as the number of demonstrations increases, even beyond the optimal pseudo-demonstration size for Naive-SemiICL. This suggests that the performance could be primarily attributed to 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. Interestingly,

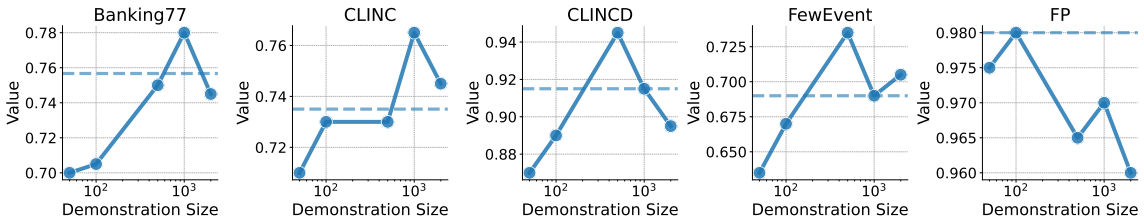

Figure 5: Performance scaling with respect to pseudo-demonstration size under IterPSD. Blue horizontal dashed line represents the best Naive-SemiICL performance on the same dataset. Results obtained from GPT-4o-mini.

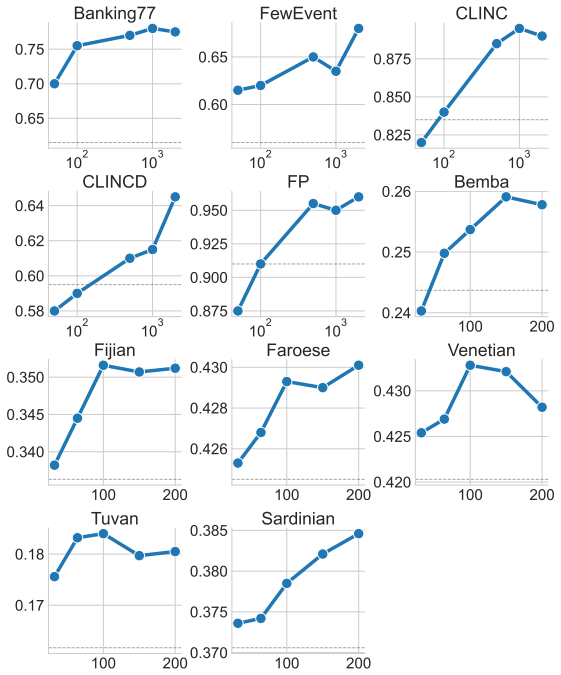

Figure 6: Scaling trend of Naive-SemiICL with no initial ground truth data. Grey dash line represents the prediction performance of zero-shot prompting. All results obtained from GPT-4o-mini.

we sample pseudo-demonstrations above a confidence threshold uniformly randomly in Naive-SemiICL, which implies that error is not introduced at a higher rate when more pseudo-demonstrations are incorporated. This indicates a previously uncharacterized phenomenon: the LLM becomes increasingly sensitive to noisy demonstrations as context length grows.

4.3 IterPSD Improves Upon Naive-SemiICL

IterPSD outperforms Naive-SemiICL across five classification tasks, as shown in Table 4. We evaluate both methods using Verbalized Confidence and Self-Consistency using 16 ground truth demonstrations. 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 scaling behavior with respect to the number of pseudo-demonstrations used in IterPSD. We observe clear scaling trends in four out of five tasks, shown in Figure 5. 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.

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.

5 Related Work

Self-Generated Demonstrations. LLM's zero-shot predictions (Kojima et al., 2022; Zou et al., 2025a) have proven to be effective sources of demonstration for in-context learning. 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 a 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 and 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-

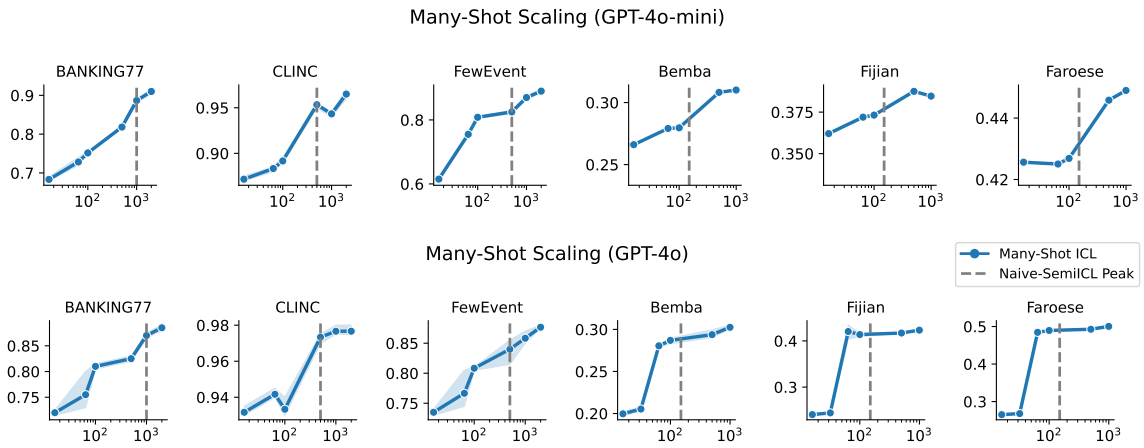

Figure 7: Many-shot scaling performance of GPT-4o-mini (top) and GPT-4o (bottom) across six selected datasets. The dashed vertical lines mark the peak performance of Naive-SemiICL.

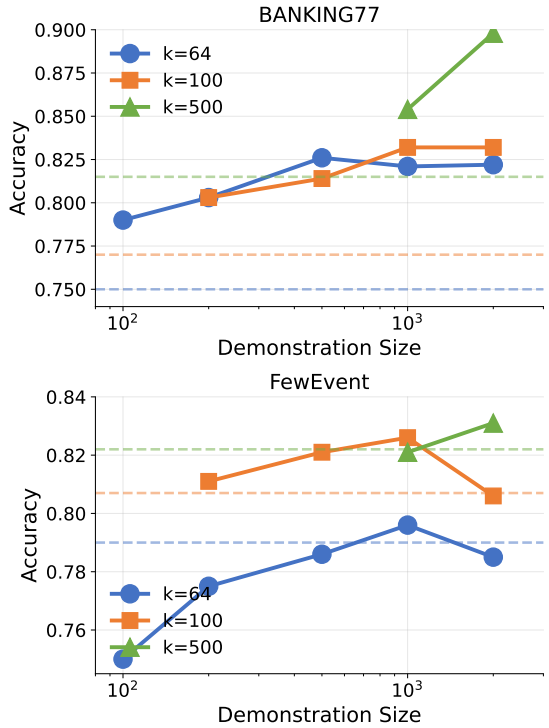

Figure 8: Naive-SemiICL across different ground truth budgets on BANKING77 & FewEvent.

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 scaling behavior between the number of examples in the demonstration and ICL performance. Our method hinges on this ability, as our proposed method, Naive-SemiICL, includes at least 64 examples in the prompt. We report similar scaling behavior 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 and 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.

6 Conclusion

We introduced a semi-supervised ICL framework that scales in-context learning performance using self-generated annotations. By analyzing both Naive-SemiICL and IterPSD, we observe that performance continues to improve with thousands of pseudo-demonstrations, revealing a surprising scaling trend in semi-supervised ICL. Naive-SemiICL demonstrates consistent gains over standard k-shot ICL across a wide range of ground truth budgets, highlighting its robustness. We further incorporate curriculum learning into IterPSD, enabling iterative refinement of pseudo-demonstrations. This leads to substantial performance improvements over Naive-SemiICL on classification tasks, validating the benefit of curriculum learning in our framework.

7 Limitation

We observed limited scaling capability of semi-supervised ICL on reasoning and translation tasks. With the advent of long-context large language models such as the Gemini family (et al., 2025), we hope that future work could break this limitation by experimenting with models that scale better with long contexts. A direction for future work, motivated by semi-supervised learning, is re-annotation during the pseudo-demonstration annotation phase of our framework. Traditional semi-supervised learning can increase the annotation quality by re-annotating those with lower confidence, which we found not to be true under the ICL setting. Solving this hindrance could be valuable.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie C.Y. Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-shot in-context learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2019. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. Self-icl: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15651–15662. Association for Computational Linguistics.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. 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, page 151–159, New York, NY, USA. Association for Computing Machinery.

Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. 2022. Torchmetrics-measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101.

Gheorghe Comanici et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Liancheng Fang, Aiwei Liu, Hengrui Zhang, Henry Peng Zou, Weizhi Zhang, and Philip S Yu. 2025. Tabgen-icl: Residual-aware in-context example selection for tabular data generation. *arXiv preprint arXiv:2502.16414*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. 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. Curran Associates Inc.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. 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)*.

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

Xiaonan Li and Xipeng Qiu. 2023. MoT: Memory-of-thought enables ChatGPT to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374, Singapore. Association for Computational Linguistics.

Yaoyiran Li, Anna Korhonen, and Ivan Vulić. 2024a. Self-augmented in-context learning for unsupervised word translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 743–753, Bangkok, Thailand. Association for Computational Linguistics.

Yaoyiran Li, Anna Korhonen, and Ivan Vulić. 2024b. Self-augmented in-context learning for unsupervised word translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 743–753, Bangkok, Thailand. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Pekka Malo, Ankur Sinha, Pekka J. Korhonen, Jyrki Wallenius, and Pyry Takala. 2013. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Sepideh Mamooler, Syrielle Montariol, Alexander Mathis, and Antoine Bosselut. 2024. Picle: Pseudo-annotations for in-context learning in low-resource named entity detection. *CoRR*, abs/2412.11923.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum learning: A survey. *CoRR*, abs/2101.10382.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-free LLM benchmark. In *The Thirteenth International Conference on Learning Representations*.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Henry Zou and Cornelia Caragea. 2023. JointMatch: A unified approach for diverse and collaborative pseudo-labeling to semi-supervised text classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7290–7301, Singapore. Association for Computational Linguistics.

Henry Zou, Yue Zhou, Weizhi Zhang, and Cornelia Caragea. 2023a. DeCrisisMB: Debiased semi-supervised learning for crisis tweet classification via memory bank. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6104–6115, Singapore. Association for Computational Linguistics.

Henry Peng Zou, Cornelia Caragea, Yue Zhou, and Doina Caragea. 2023b. Semi-supervised few-shot learning for fine-grained disaster tweet classification. In *Proceedings of the 20th International ISCRAM Conference*. ISCRAM 2023.

Henry Peng Zou, Zhengyao Gu, Yue Zhou, Yankai Chen, Weizhi Zhang, Liancheng Fang, Yibo Wang, Yangning Li, Kay Liu, and Philip S Yu. 2025a. Testnuc: Enhancing test-time computing approaches through neighboring unlabeled data consistency. *arXiv preprint arXiv:2502.19163*.

Henry Peng Zou, Siffi Singh, Yi Nian, Jianfeng He, Jason Cai, Saab Mansour, and Hang Su. 2025b. Glean: Generalized category discovery with diverse and quality-enhanced llm feedback. *arXiv preprint arXiv:2502.18414*.

A Prompts

The prompts are illustrated in Table 8. {CAPITAL LETTERS} enclosed in curly brackets are variables that are substituted during inference.

B Experimental Details

B.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 fewer 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.2 Computational Budget

We ran all of our experiments on an Apple M3 chip, where embedding-based search constitutes less than 1% of the computation time during IterPSD. The embeddings can be precomputed during data processing for each dataset, as they only need to be computed once. It took about 400ms to retrieve each embedding from the OpenAI API. The cost of generating the embeddings is \$0.13/million tokens. The average cost across experimental settings is 6.84 USD, with the largest experiment costing 24.31 USD.

B.3 Dataset Details

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 involve 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.

B.4 Baseline Details

We observe that ground-truth few-shot baselines (Section 3) do not necessarily scale with more ground-truth demonstrations. Thus, we report the best performing k-shot baseline where $k \leq 16$, which we report in Table 6.

For MoT (Li and Qiu, 2023), we follow a recommended configuration of 5 clusters. For retrieval, we employ the same text embeddings, `text-embedding-3-large` as Naive-SemiICL, and the same confidence threshold set at the 90th percentile. Since MoT needs to query the LLM k times to select

		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 5: 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.

	GPQA	LiveBench Math	Logical7	Geometric Shapes	Date
GPT-4o-mini	4	0	8	8	16
GPT-4o	4	0	8	8	16

Table 6: Best number of shots for the baseline on reasoning tasks. We use the same number of pseudo-demonstrations for Naive-SemiICL.

the most relevant examples, it is not suitable for classification and translation tasks that might utilize many examples, we only compare MoT to Naive-SemiICL on reasoning tasks.

B.5 Applying Confidence Scores

On all tasks, we sample from the LLMs 10 times to compute the Self-Consistency score. Self-consistency is unsuitable for translation tasks due to the computational challenges of assessing equivalence between translations. Instead, we introduce Back-Translation, which evaluates translation quality by translating the output back to the original language. The confidence score is then derived using the cosine similarity (on embeddings) between the back-translation and the original input. A detailed description of Back-Translation is provided in Appendix B.6.

B.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

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

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 $\hat{s}$

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

where ϕ is an embedding function.

B.7 Hyperparameters

Confidence Threshold. We find that setting the confidence threshold λ at the 90th percentile of all generate pseudo-demonstrations $\mathcal{D}_{\mathrm{PSD}}$ allowed Naive-SemiICL to achieve competitive performance. We assume this threshold in this work unless stated otherwise.

IterPSD takes 3 hyperparameters: chunk size K, which controls how many data to annotate in each iteration, ϵ, which controls the proportion of random sample in each chunk, and κ, the maximum number of examples allowed in the demonstration while generating pseudo-demonstrations. We experiment with $K \in \{100, 500, 1000\}$, $\epsilon \in \{0.5, 0.8, 1.0\}$, $\kappa \in \{300, 500, 1000\}$. We find that $\epsilon = 0.8$, $K = 500$, and $\kappa = 1000$ yielded the best results on all tasks, except on FP, where $K = 100$ and $\kappa = 300$ yielded the best result.

C Effects of Different Confidence Methods

In this section, we examine the performance of Naive-SemiICL paired with different confidence methods, which we compile as Table 5. 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.

Model	Bemba	Fijian	Faroese	Venetian	Tuvan	Sardinian	Banking	FewEvent	CLINC	CLINCD	FP
4o-mini	100	150	150	100	64	64	1000	2000	2000	2000	500
4o	100	150	100	200	100	100	1000	500	500	1000	100

Table 7: Optimal demonstration counts for Naive-SemiICL per dataset under 4o-mini and 4o models.

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: $K_{\text{random}} = \epsilon K, K_{\text{sim}} = (1-\epsilon)K$;
4: Compute $d_{ij} = \text{sim}_{\cos}(\phi(x_i), \phi(x_j))$ 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: $S_{\text{sim}} = \{x_j | d_j \in \text{Smallest}_{K_{\text{sim}}}\{d_j\}\}$;
{*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 $\overline{\mathcal{D}}_l - S_{\text{sim}}$);
8: $S = S_{\text{sim}} \cup S_{\text{random}}$;
9: **Return** S;

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 falls 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.

D Sampler for IterPSD

In our IterPSD experiments, we implement Algorithm 3 for the ϵ-random sampling strategy in Section 2.3. In Line 4 and 5, the algorithm computes the similarity between every pair of annotated and unannotated pairs. Then the closest K_{sim} unannotated examples are chosen (Line 6). The remaining K_{random} examples are sampled randomly from those not chosen. Lastly, the similar and random samples are combined and returned.

A way of sampling diverse random samples is the clustering strategy in (Zhang et al., 2023). In this method, samples are divided into clusters according to some clustering algorithm. Then k_{random} most representative samples are from each cluster. Representativeness is computed by computing the shortest sum of euclidean distance to other examples in the cluster.

Table 8: 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>