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

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 in zero-shot, few-shot, and many-shot settings. Notably, we observe a *scaling law* with this baseline, where optimal performance is achieved with more than 1,000 demonstrations. To fully exploit the many-shot capabilities of semi-supervised ICL, we introduce IterPSD, an iterative 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. Code is available at: https://anonymous.4open.science/r/semi-supervised-icl-FA07

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 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, (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:

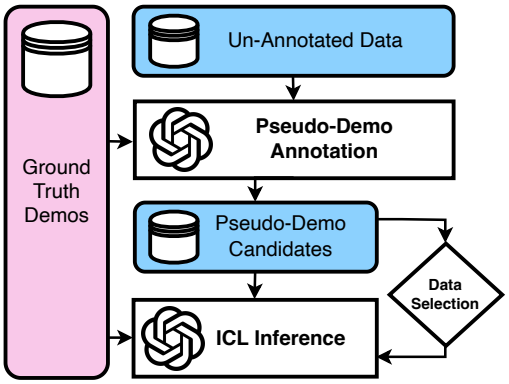

Figure 1: Semi-supervised ICL Framework. Ground truth data are used as demonstration for generating pseudo-demonstrations from unannotated data. The generated pseudo-demonstrations conjunctively with a small ground truth demonstration, are selectively used as demonstrations for the final prompting.

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 3).

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 for each input:

$$(y, r, c) = \text{LM}(\rho_{\mathcal{T}}, \mathcal{E}, x) \tag{1}$$

Like traditional ICL, the LLM is prompted with a task instruction $\rho_{\mathcal{T}}$ associated with task $\mathcal{T}$, a set of demonstrations $\mathcal{E}$, and an input x. Unlike traditional ICL, however, the model additionally returns a confidence score c along with the predicted output y and rationale r[1], providing a measure of certainty for its predictions. We discuss the specific choice of confidence measure in Section 3.

Semi-Supervised ICL. Beyond ground-truth annotations, Semi-Supervised ICL leverages unannotated data to enrich demonstrations. The setting assumes the availability of a *ground-truth dataset* $\mathcal{D}_g = \{(x_i, y_i)\}^{N_l}$, usually small in quantity, alongside a large pool of *unannotated data* $\mathcal{X}_u = {x_i}^{N_u}$. Semi-Supervised ICL augments the limited pool of ground-truth examples by generating *pseudo-demonstrations* $\mathcal{D}_{\text{PSD}}$ from the unannotated data. Formally,

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

where $(\tilde{y}, \tilde{r}, \tilde{c}) = \text{LM}(\rho_{\mathcal{T}}, \mathcal{E}_g, x)$ are generated by the LLM in an ICL fashion using a set of ground-truth demonstrated ns $\mathcal{E}_g \subseteq \mathcal{D}_g$. Low-confidence examples are filtered out according to a confidence threshold λ:

$$\mathcal{D}^{\lambda}_{\text{PSD}} = \{(x, \tilde{r}, \tilde{y}, \tilde{c}) | \tilde{c} \geq \lambda, (x, \tilde{r}, \tilde{y}, \tilde{c}) \in \mathcal{D}_{\text{PSL}}\}. \tag{3}$$

During Semi-Supervised ICL inference, we sample pseudo-demonstrations $\mathcal{E}_{\text{PSD}}$ from the filtered set $\mathcal{D}^{\lambda}_{\text{PSD}}$, which we detail the specific methods in Appendix B.6. The LLM is then prompted with these pseudo-demonstrations alongside the ground-truth demonstrations from which they were generated:

$$(\hat{y}, \hat{r}, \hat{c}) = \text{LM}(\rho_{\mathcal{T}}, \mathcal{E}_l \cup \mathcal{D}^{\lambda}_{\text{PSD}}, x). \tag{4}$$

Most of the internal mechanisms of Semi-Supervised ICL are encapsulated by Equation 2, where pseudo-demonstrations are generated, while Equation 3 and Equation 4 represent simple operations. Next, we introduce two approaches for generating pseudo-demonstrations.

2.2 A Simple Semi-Supervised ICL Baseline

We propose a simple method that generates pseudo-demonstrations in a single iteration (Algorithm 1). We dub this method, along with the rest of the Semi-Supervised ICL framework, **Naive-SemiICL**. The method simply iterates over the unlabeled data for one iteration and generates a prediction, a rationale, and a confidence score for each input. As

[1] In practice, generating rationales is optional. For example, one can query the LLM to directly generate the answer to a mathematical problem without intermediate reasoning steps.

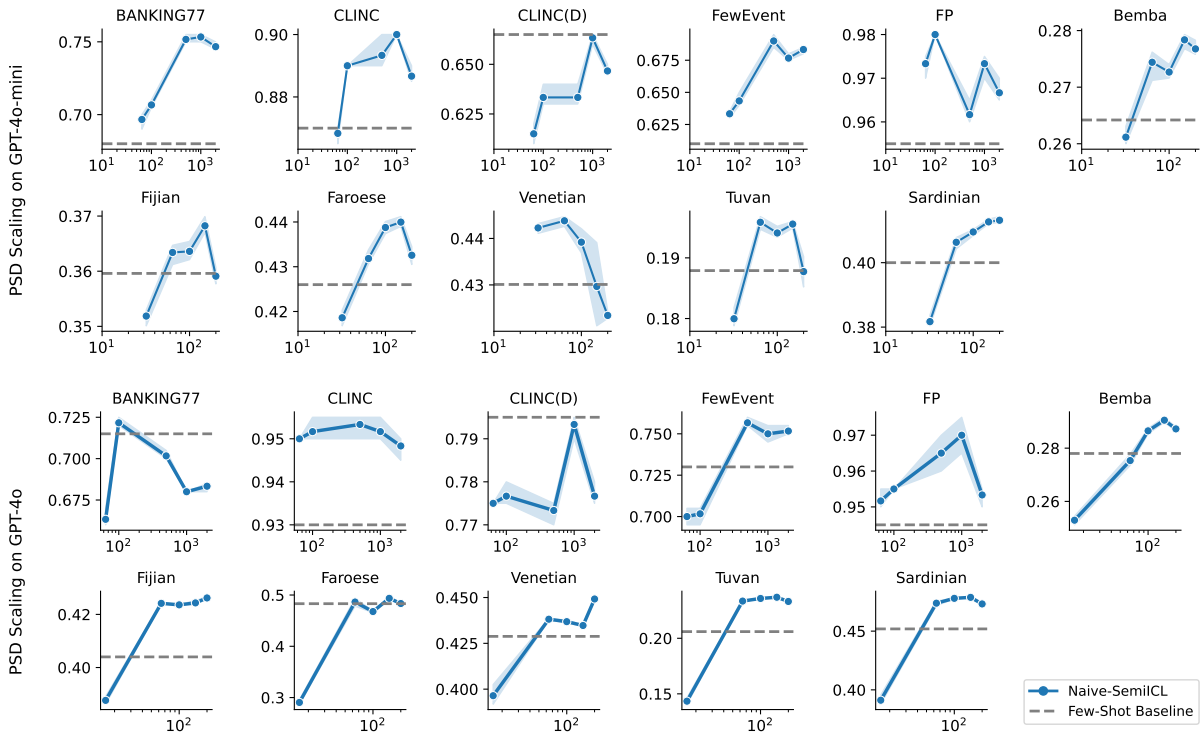

Figure 2: Scaling trend of Naive-SemiICL on classification and translation tasks with GPT-4o and GPT-4o-mini. The dashed gray line represents the few-shot baseline. Both model exhibits a scaling trend on most tasks. All experiments are performed with a ground truth budget of $k_l = 16$.

Algorithm 1 Naive-SemiICL.

1: **Input:** prompt $\rho_{\mathcal{T}}$, ground-truth demonstrations $\mathcal{E}_l \subseteq \mathcal{D}_l$, confidence score $\mathcal{C}$;
2: Initialize $\mathcal{D}_{\text{PSD}} = \emptyset$;
3: **for** $x \in \mathcal{X}_u$ **do**
4: $\quad \tilde{y}, \tilde{r}, \tilde{c} = \text{LM}(\rho_{\mathcal{T}}, \mathcal{E}_l, x)$;
5: $\quad \mathcal{D}_{\text{PSD}} = \mathcal{D}_{\text{PSD}} \cup \{(x, \tilde{r}, \tilde{y}, \tilde{c})\}$;
6: **end for**
7: **Return** $\mathcal{D}_{\text{PSD}}$;

the simplest form of Semi-Supervised ICL, it provides effective in-context learning signals by filtering out low-quality pseudo-demonstrations. We experiment with three commonly used confidence measures on 16 datasets spanning 9 tasks, and show that this simple baseline consistently outperforms a strong 16-shot ICL baseline (Section 4.1).

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 own prompt until reaching the maximum number of allowed demonstrations (Line 6), 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 12).

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 caused by long context lengths, 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 the generated pseudo-demonstrations, ensuring that only high-quality examples are retained (Line 7).

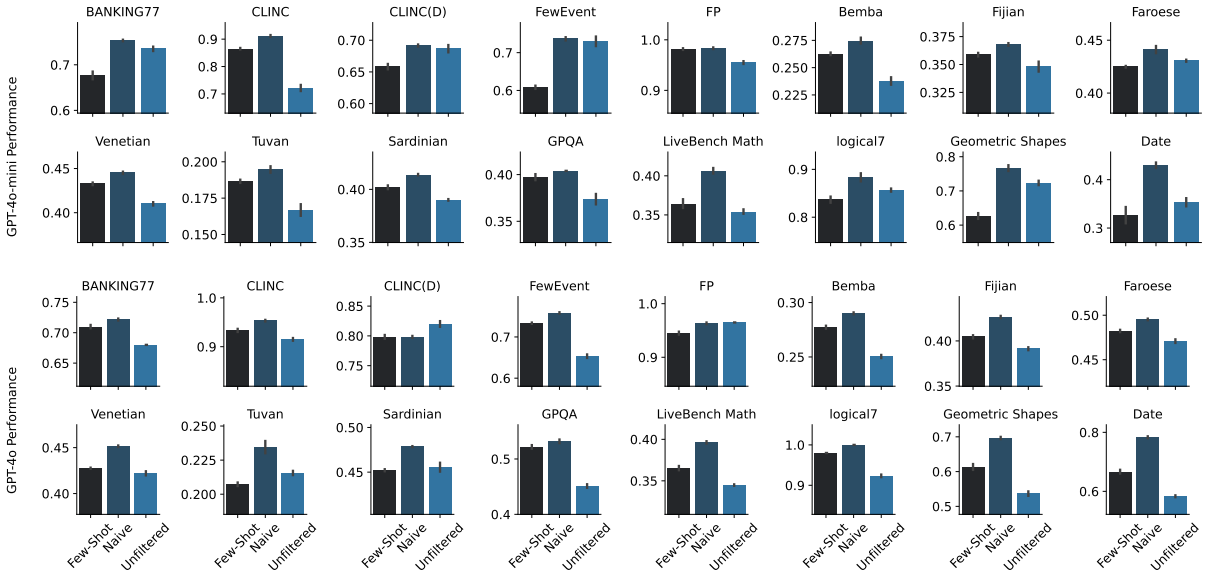

Figure 3: Comparison of GPT-4o-mini (top) and GPT-4o (bottom) performance across multiple datasets using three different methods: Few-Shot, Naive-SemiICL (Naive), and Naive-SemiICL without filtering (Unfiltered).

Order of Annotation. To enhance the accuracy of pseudo-demonstration annotations, we introduce the ϵ-Random Sampler (Algorithm 3), a sampling strategy that selects both similar and diverse examples from the unannotated pool. At each iteration, a proportion $(1-\epsilon)$ of examples is chosen based on their cosine similarity to the nearest previously annotated instances (Line 6)[2], ensuring that each selected example is similar to an existing annotation. The rest of the examples are chosen diversely in a clustering fashion similar to the one in (Zhang et al., 2023). This approach that considers both the similarity and diversity aligns with curriculum learning (Soviany et al., 2021) in semi-supervised learning, which facilitate self-training by balancing a mixture of confident and uncertain predictions.

Mitigating Confirmation Bias. To maintain annotation quality, we find that at least half of the data ($\epsilon \geq 0.5$) should be sampled diversely (Line 7)[3]. When $\epsilon = 0$, selections are exclusively based on similarity to previously annotated examples. 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), highlighting a strong connection between Semi-Supervised ICL and traditional semi-supervised learning frameworks.

[2]We compute cosine similarity using vector embeddings generated by OpenAI's *text-embedding-3-large*.

[3]We found the best performing ϵ to be 0.8 in most of our experiments.

3 Experimetnal 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 (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 (Costa-Jussà et al., 2022): Bemba, Fijian, Faroese, Tuvan, Venetian, and Sardinian.
- **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

Algorithm 2 IterPSD

Input: prompt $\rho_{\mathcal{T}}$, ground-truth demonstrations $\mathcal{E}_l \subseteq \mathcal{D}_l$, chunk size K, ratio of random examples ϵ, maximum number of pseudo-demonstrations κ, confidence score $\mathcal{C}$;
Initialize $\mathcal{D}_{\text{PSD}} = \emptyset$; {*Set of all the annotated pseudo-demonstrations.*}
Initialize $\mathcal{D}^{\lambda}_{\text{PSD}} = \emptyset$
Initialize $\overline{\mathcal{D}}_{\text{PSD}} = \mathcal{X}_u$; {*Set of un-annotated data yet to be annotated.*}
while $\overline{\mathcal{D}}_{\text{PSD}} \neq \emptyset$ **do**
  if $|\mathcal{D}^{\lambda}_{\text{PSD}}| > \kappa$ **then**
    $\mathcal{D}^{\lambda}_{\text{PSD}}$ = top-κ confident examples in $\mathcal{D}_{\text{PSD}}$;
  end if
  {*Cap the demonstration at a maximum size, prevent performance degradation from long-context.*}
  $S = \text{Sampler}(\mathcal{E}_{\text{Iter}}, \overline{\mathcal{D}}_{\text{PSD}}, K, \epsilon)$;
  {*Retrieves a sample of size K using ϵ-Random Sampler*}
  $\mathcal{D}_S = \text{Naive-SemiICL}(S, \rho_{\mathcal{T}}, \mathcal{E}_l \cup \mathcal{D}^{\lambda}_{\text{PSD}})$;
  {*One iteration of Naive-SemiICL.*}
  $\mathcal{D}^{\lambda}_S = \{(x, \tilde{r}, \tilde{y}, \tilde{c}) | (x, \tilde{r}, \tilde{y}, \tilde{c}) \in \mathcal{D}_S, \tilde{c} \geq \lambda\}$; {*Filter by confidence.*}
  $\mathcal{D}^{\lambda}_{\text{PSD}} = \mathcal{D}^{\lambda}_{\text{PSD}} \cup \mathcal{D}^{\lambda}_S$;
  $\mathcal{D}_{\text{PSD}} = \mathcal{D}_{\text{PSD}} \cup \mathcal{D}_S$;
  $\overline{\mathcal{D}}_{\text{PSD}} = \overline{\mathcal{D}}_{\text{PSD}} - \mathcal{D}_S$;
end while
Return $\mathcal{D}_{\text{PSD}}$;

Algorithm 3 ϵ-Random Sampler

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

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 for baseline results (OpenAI, 2024) (Section 4.1). The remaining results are based on a single trial.

Baselines. For baseline comparisons, we experiment with different pseudo-demonstration sizes for Naive-SemiICL: $k_u \in \{32, 64, 100, 500, 1000, 2000\}$ for classification tasks, and $k_u \in \{32, 64, 100, 150, 200\}$ for translation tasks.

- **k-Shot ICL.** The LLM is prompted with k ground truth annotated examples, where k ranges from 0 to 500. Base on the number of ground truth annotation used, we divide our experiments into zero-shot (Table 1)), few-shot (Fig. 3), and many-shot (Fig. 6) settings.

- **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. Configuration details are provided in Appendix B.4.

- **Reinforced ICL.** (Agarwal et al., 2024) demonstrate 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 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 include Auto-CoT in our experiments.

Confidence Scores. We primarily evaluate three confidence metrics: Verbalized Confidence which prompts the LLM to generate the confidence score (Table 7), Entropy, and Self-Consistency. Self-

	Task	Zero-shot	Naive	Improv.
Classification	Banking	61.50	**78.00**	**26.8%**
	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. For Naive-SemiICL, we report the best performing number of pseudo-demonstrations.

Consistency measures the confidence as the frequency of the most frequent answer, and entropy is defined as

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

Hyperparameters. Unless stated otherwise, we filter all generated pseudo-demonstrations using the confidence threshold at the 90th percentile. We discuss the hyperparameters of IterPSD in Appendix B.8.

Models. We experiment with GPT-4o-mini and GPT-4o, checkpointed on 2024-07-18 and 2024-11-20, respectively, for all of our experiments. 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. 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 Tables 5 and 6.

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.

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. A detailed breakdown of the best performance across different confidence scores is provided in Appendix C.

We highlight the effectiveness of Naive-SemiICL in extremely low-resource settings through a zero-shot experimental design. We generate pseudo-demonstrations with *no initial ground-truth demonstrations* and compare Naive-SemiICL to zero-shot prompting. 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 resource-constrained conditions.

Additionally, we found Naive-SemiICL to be effective in high-resource settings. Figure 6 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.

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:

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 3: 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.

(1) MoT uses Entropy to filter low-quality demonstrations, which we show to be less reliable than Verbalized Confidence (see Table 4); 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.

4.2 Scaling Law for Semi-Supervised ICL

We observe a scaling law for Semi-Supervised ICL, similar to the one reported in many-shot ICL (Agarwal et al., 2024), on classification and translation tasks. We illustrate this trend in Figure 2. Across all configurations, 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. GPT-4o exhibits a more stable scaling trend than GPT-4o-mini on translation tasks, with performance peaking later and declining more gradually. With more available ground-truth data, we also observe scaling trends on BANKING77 and FewEvent (Figure 6).

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 5 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.

4.3 IterPSD Improves Upon Naive-SemiICL

IterPSD outperforms Naive-SemiICL across five classification tasks, as shown in Table 3. 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 4. 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.

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. 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 and Qiu, 2023) builds on top of Auto-CoT with extra an extra step of semantic filtering. At each

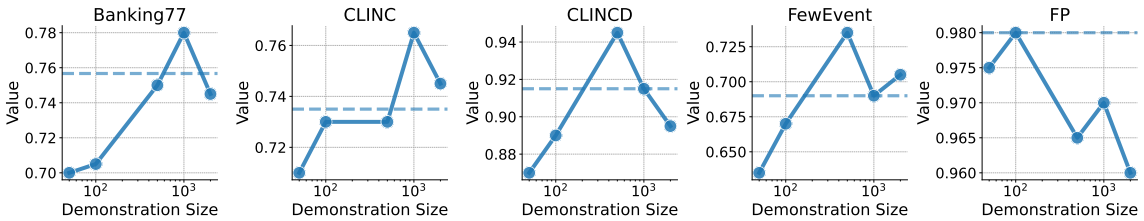

Figure 4: Scaling trend of IterPSD on five benchmark tasks. Blue horizontal dashed line represents the best performing Naive-SemiICL on the same dataset.

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 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 enhances self-generated annotations through confidence-based data selection and iterative annotation. Our analysis of Naive-SemiICL with increasing amounts of ground-truth data reveals diminishing returns—while additional ground-truth annotations improve performance, the relative contribution of pseudo-demonstrations decreases. This suggests that semi-supervised ICL is particularly effective in low-resource settings, yet remains beneficial even when more ground-truth data is available. We further identify a scaling law in semi-supervised ICL, showing that models achieve optimal performance with over 1,000 pseudo-demonstrations. Our simple semi-supervised method, Naive-SemiICL, outperforms a strong 16-shot ICL baseline, achieving an average performance gain of 9.94% across 16 datasets. We also propose IterPSD, an iterative refinement approach for pseudo-demonstrations, which yields up to 6.8% additional gains on classification tasks.

7 Limitation

While this work investigates the potential of semi-supervised ICL, several limitations remain. First, the reliance on SOTA LLMs for ICL introduces substantial computational overhead, posing challenges for researchers and practitioners with limited resources. Second, although we have shown that the incorporation of pseudo-demonstration generation strategies enhances and improves ICL performance on two models, the effectiveness of our proposed method might be sensitive to the choice of model. Future work can explore more advanced confidence calibration techniques for pseudo-demonstration selection, such as adaptive thresholding. Additionally, noise-aware in-context learning remains an under-explored domain that could potentially improve the robustness of Semi-Supervised ICL.

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*.

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.

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.

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.

OpenAI. 2024. Gpt-4o system card. *https://arxiv.org/pdf/2410.21276*.

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*.

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 7. {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 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.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 it only needs to be computed once. It took about 400ms to retrieve each embeddings from the OpenAI API. The cost of generating the embeddings is $0.13/million tokens. We ran all of our experiments on a $1,000 budget.

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 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

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

Table 4: 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.

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.

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

Table 5: Best number of shots for the baseline on reasoning tasks. We use the same number of shots for Naive-SemiICL.

B.4 Baseline Details

We observe that few-shot baselines (Section 3) does not necessarily scale with more demonstrations. Thus, we report the best performing k-shot baseline where $k \leq 16$, which we report in Table 5.

We experiment with different pseudo-demonstration sizes k_u for Naive-SemiICL: $k_u \in \{32, 64, 100, 150, 200\}$ on classification tasks, and $k_u \in \{32, 64, 100, 150, 200\}$ on translation and reasoning tasks.

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 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.7.

B.6 Pseudo-Demonstration Sampling

On classification tasks, BigBenchHard tasks and GPQA, we sample diversely by evenly sample predicted labels. For translation tasks and LiveBench Math, we utilize OpenAI's `text-embedding-3-large` to generate embeddings for each example input. Then we cluster the embeddings into clusters and evenly sample the most representative (the one closest to the cluster centroid) instances from each cluster.

B.7 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

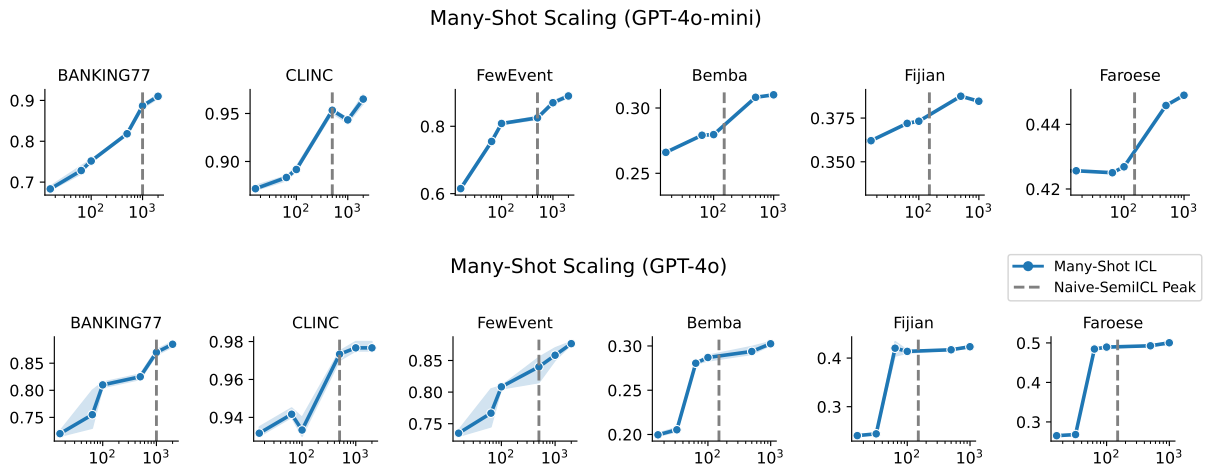

Figure 5: Many-shot scaling performance of GPT-4o-mini (top) and GPT-4o (bottom) across six selected datasets. The x-axis represents the number of shots (log scale), and the y-axis represents performance. The solid blue lines indicate many-shot in-context learning (ICL), while the dashed vertical lines mark the peak performance of Naive-SemiICL. Both models scale beyond the peak the performance of pseudo-demonstration approach.

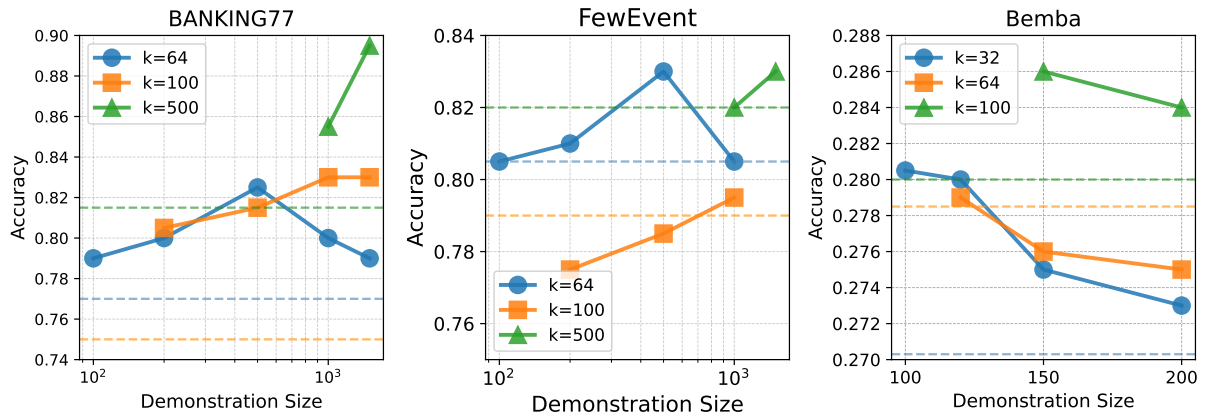

Figure 6: We compare Naive-SemiICL accuracy across different ground truth demonstration sizes, with baseline performances indicated by dashed lines. On FewEvent, the maximum number of pseudo-demonstrations is capped at 1000 due to the limited availability of pseudo-demonstrations after filtering.

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.8 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, κ, the maximum amount 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.

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 6: Demonstration counts per dataset for 4o-mini and 4o models

C Effects of Different Confidence Methods

In this section, we examine the performance Naive-SemiICL paired with different confidence methods, which we compile as Table 4. 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.

Table 7: 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>