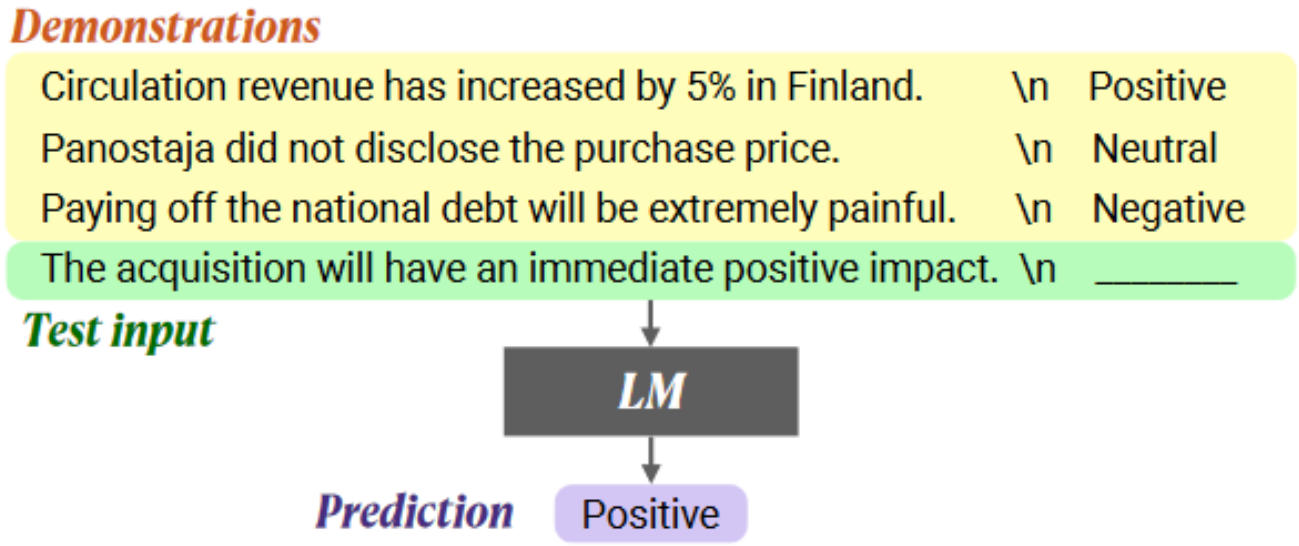
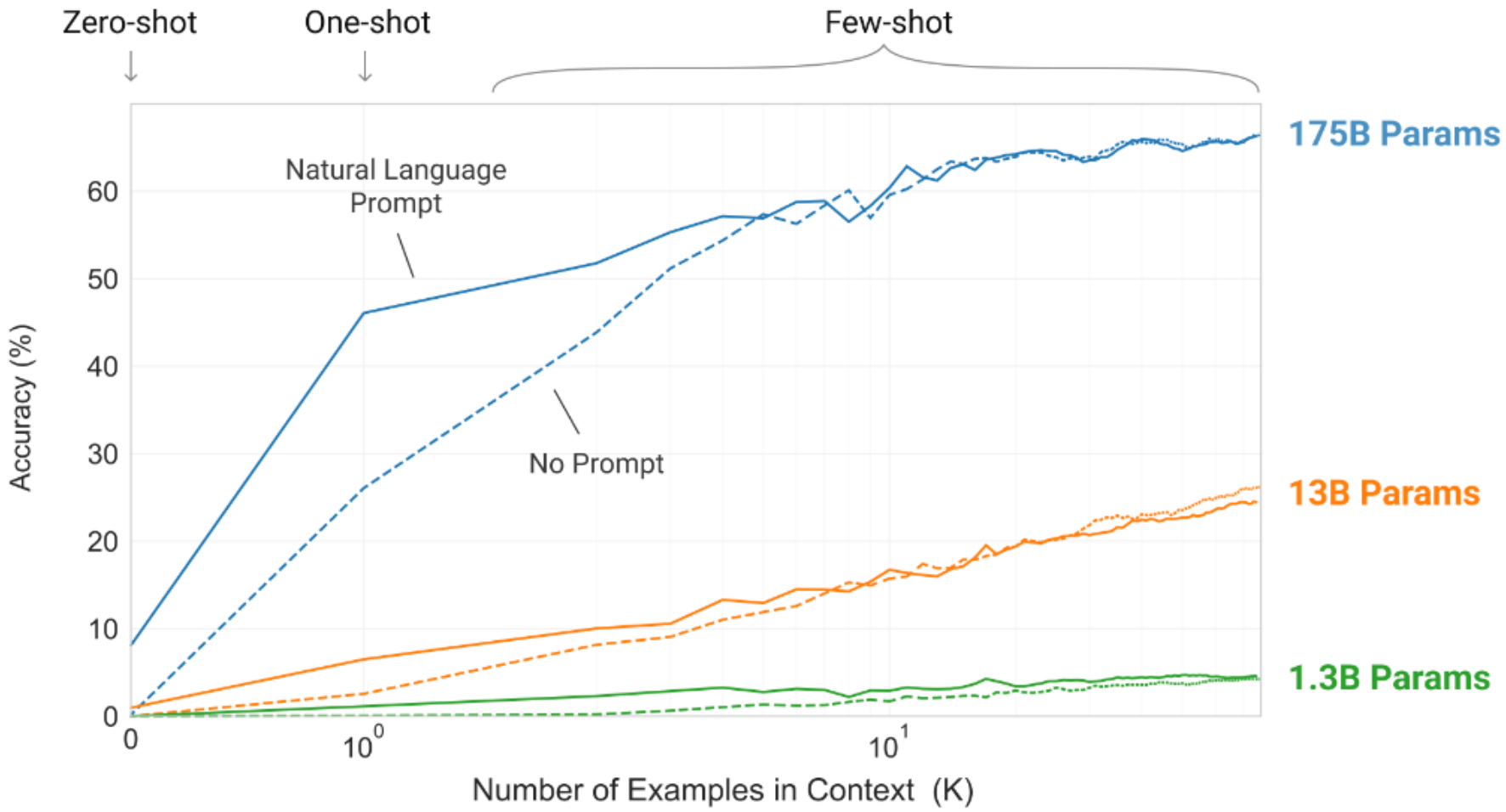
In-Context Learning

What is in-context learning?

Using the text input of a pretrained language model as a form of task specification: the model is conditioned on a natural language instruction and/or a few demonstrations of the task and is then expected to complete further instances of the task simply by predicting what comes next.





Larger models make increasingly efficient use of in-context information.
We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task. The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

Sewon Min[1,2] **Xinxi Lyu**[1] **Ari Holtzman**[1] **Mikel Artetxe**[2]
Mike Lewis[2] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon,alrope,ahai,hannaneh,lsz}@cs.washington.edu
{artetxe,mikelewis}@meta.com

Motivation

- Large language models (LMs) are able to in-context learn—perform a new task via inference alone by conditioning on a few input-label pairs (demonstrations) and making predictions for new inputs.
- However, there has been little understanding of *how* the model learns and *which* aspects of the demonstrations contribute to end task performance.
- We show that ground truth demonstrations are in fact not required—randomly replacing labels in the demonstrations barely hurts performance.

Introduction

- In this paper, we show that ground truth demonstrations are in fact not required for effective in-context learning. Specifically, replacing the labels in demonstrations with random labels barely hurts performance in a range of classification and multi-choice tasks.

Introduction

- In summary, our analysis provides a new way of understanding the role of the demonstrations in in-context learning.
- We empirically show that the model counter-intuitively does not rely on the ground truth input-label mapping provided in the demonstrations as much as we thought.
- The model nonetheless still benefits from knowing the label space and the distribution of inputs specified by the demonstrations. We also include a discussion of broader implications, e.g., what we can say about the model learning at test time, and avenues for future work.

Experimental Setup

- **Models**

We experiment with 12 models in total. We include 6 language models , all of which are decoder-only, dense LMs. We use each LM with two inference methods, direct and channel.

Model	# Params	Public	Meta-trained
GPT-2 Large	774M	✓	✗
MetaICL	774M	✓	✓
GPT-J	6B	✓	✗
fairseq 6.7B†	6.7B	✓	✗
fairseq 13B†	13B	✓	✗
GPT-3	175B‡	✗	✗

Experimental Setup

- **Evaluation Data**

We evaluate on 26 datasets, including sentiment analysis, paraphrase detection, natural language inference, hate speech detection, question answering, and sentence completion. All datasets are classification and multi-choice tasks.

- **Other Details**

We use k=16 examples as demonstrations by default for all experiments in the paper, unless otherwise specified. Examples are sampled at uniform from the training data. We choose a set of k training examples using 5 different random seeds and run experiments 5 times. We use the minimal templates in forming an input sequence from an example.

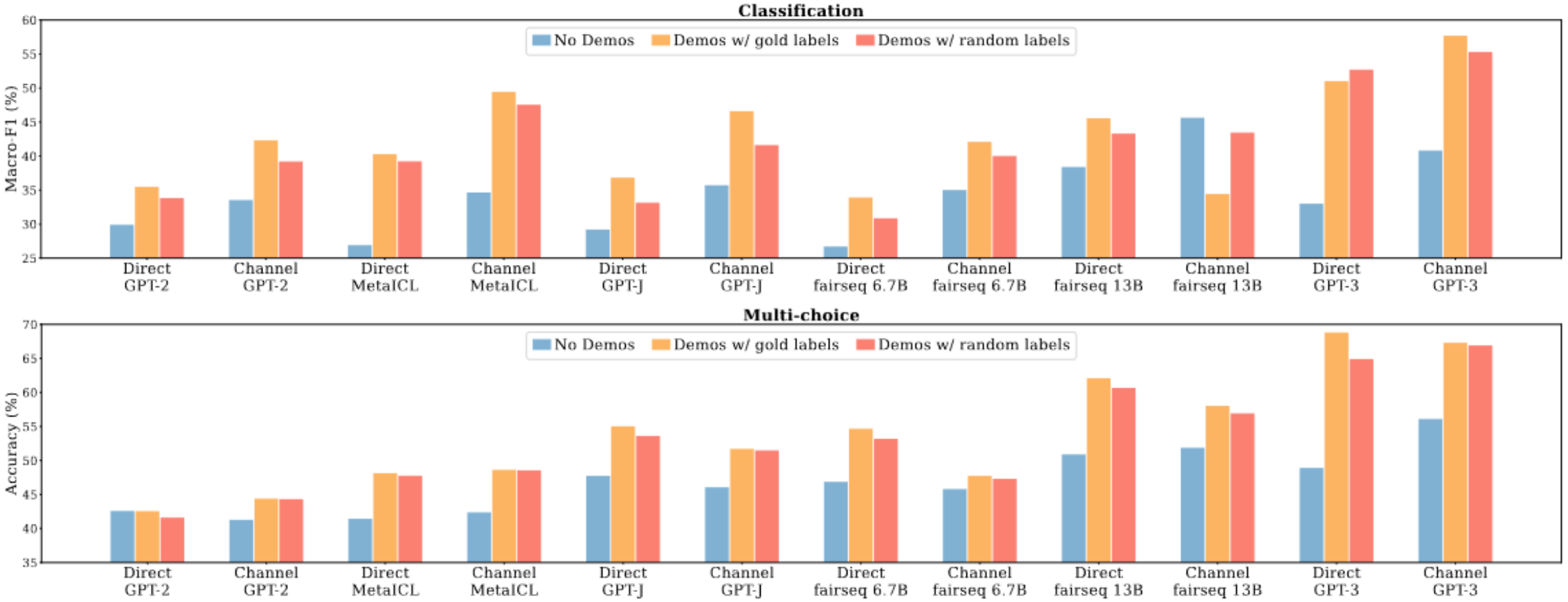
Ground Truth Matters Little

- To see the impact of correctly-paired inputs and labels in the demonstrations—which we call the ground truth input-label mapping—we compare the following three methods.
- **No demonstrations** is a typical zero-shot method that does not use any labeled data. A prediction is made via $\text{argmax}_{y \in C} P(y|x)$, where x is the test input and C is a small discrete set of possible labels.
- **Demonstrations w/ gold labels** are used in a typical in-context learning method with k labeled examples $(x_1, y_1)...(x_k, y_k)$. A concatenation of k input-label pairs is used to make a prediction via $\text{argmax}_{y \in C} P(y|x_1, y_1 ... x_k, y_k, x)$.

Ground Truth Matters Little

- **Demonstrations w/ random labels** are formed with random labels, instead of gold labels from the labeled data. Each xi ($1 \leq i \leq k$) is paired with $\tilde{y}_i$ that is randomlv sampled at uniform from C. A concatenation of $(x_1, \tilde{y}_1)...(x_k, \tilde{y}_k)$ is then used to make a prediction via $\text{argmax}_{y \in C} P(y|x_1, \tilde{y}_1 ... x_k, \tilde{y}_k, x)$.

Ground Truth Matters Little



We then find that replacing gold labels with random labels only marginally hurts performance.

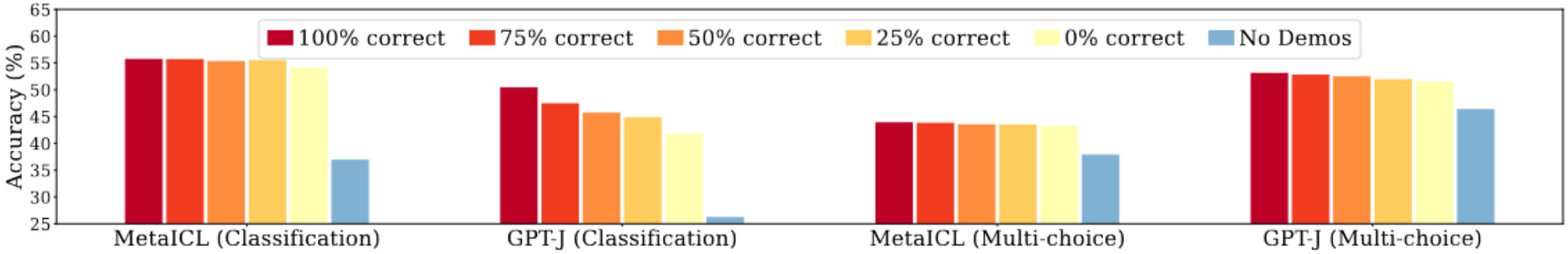
Ground Truth Matters Little

This result indicates that the ground truth input-label pairs are not necessary to achieve performance gains. This is counter-intuitive, given that correctly paired training data is critical in typical supervised training—it informs the model of the expected input-label correspondence required to perform the downstream task. Nonetheless, the models do achieve non-trivial performance on the downstream tasks. This strongly suggests that **the models are capable of recovering the expected input-label correspondence for the task; however, it is not directly from the pairings in the demonstrations.**

Ablations

- **Does the number of correct labels matter?**

To further examine the impact of correctness of labels in the demonstrations, we conduct an ablation study by varying the number of correct labels in the demonstrations.

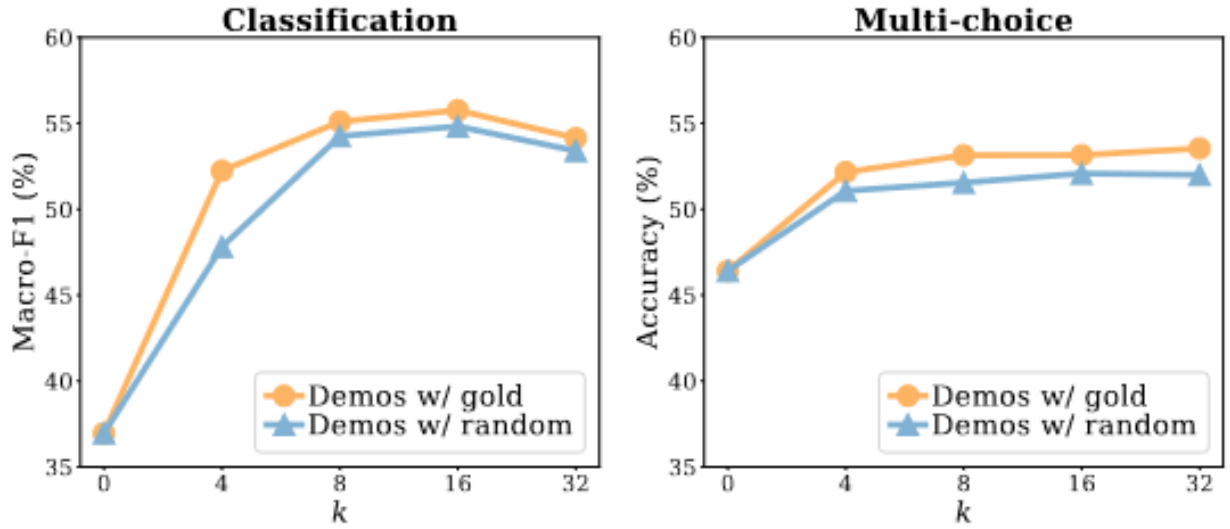


Ablations

- **Is the result consistent with varying k?**

We study the impact of the number of input-label pairs (k) in the demonstrations.

We hypothesize that larger labeled data is beneficial mainly for supervising the input-label correspondence, and other components of the data like the example inputs, example labels and the data format are easier to recover from the small data, which is potentially a reason for minimal performance gains from larger k.

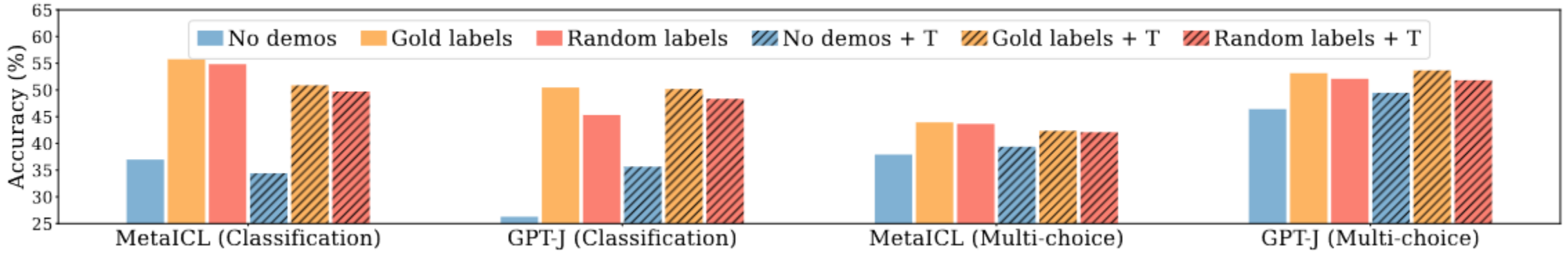


Ablations

- **Is the result consistent with better templates?**

While we use minimal templates by default, we also explore manual templates, i.e., templates that are manually written in a dataset-specific manner, taken from prior work.

Figure shows that the trend—replacing gold labels with random labels barely hurting performance—holds with manual templates.

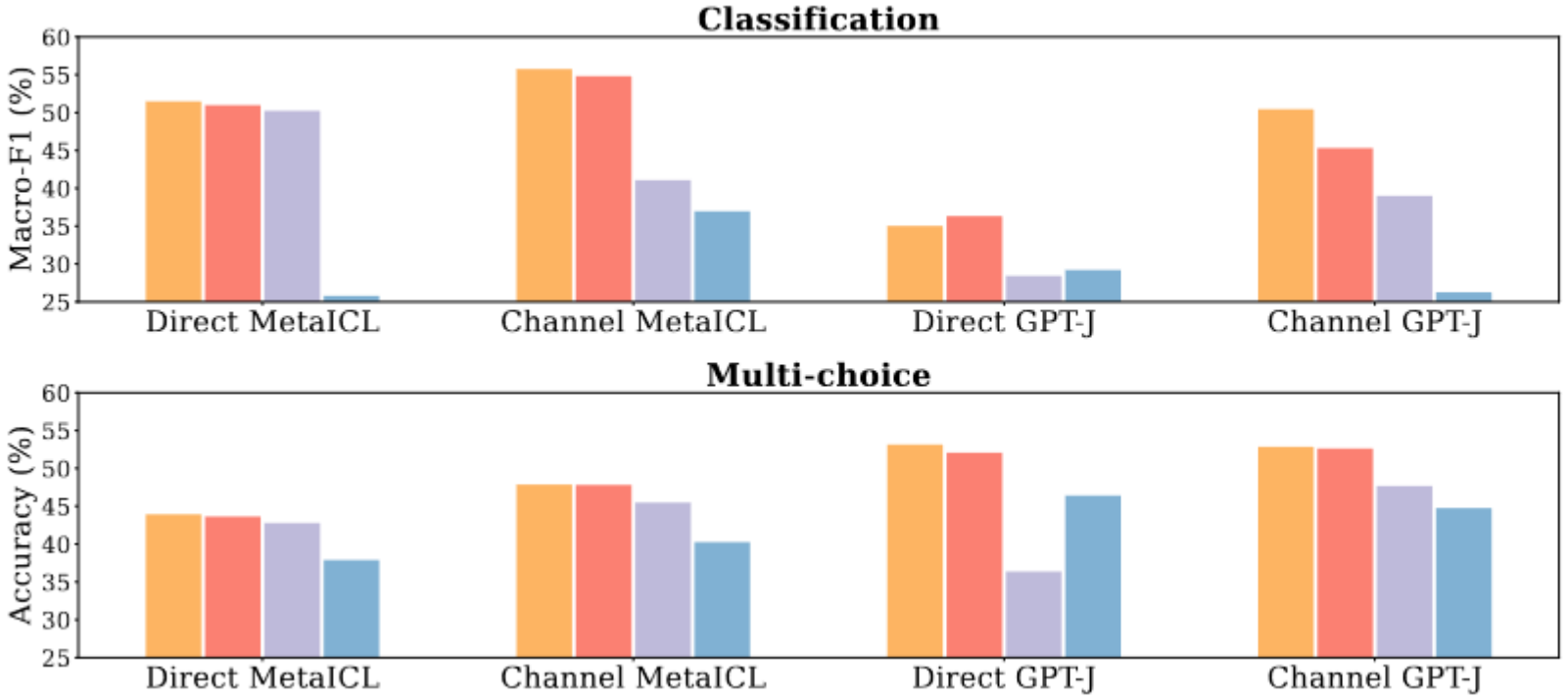


Why does In-Context Learning work?

We identify four aspects of the demonstrations $(x_1, y_1)...(x_k, y_k)$ that potentially provide learning signal.

- The input-label mapping, i.e., whether each input x_i is paired with a correct label y_i.
- The distribution of the input text, i.e., the underlying distribution that $x_1...x_k$ are from.
- The label space, i.e., the space covered by $y_1 \cdots y_k$.
- The format—specifically, the use of input-label pairing as the format.

Impact of the distribution of the input text

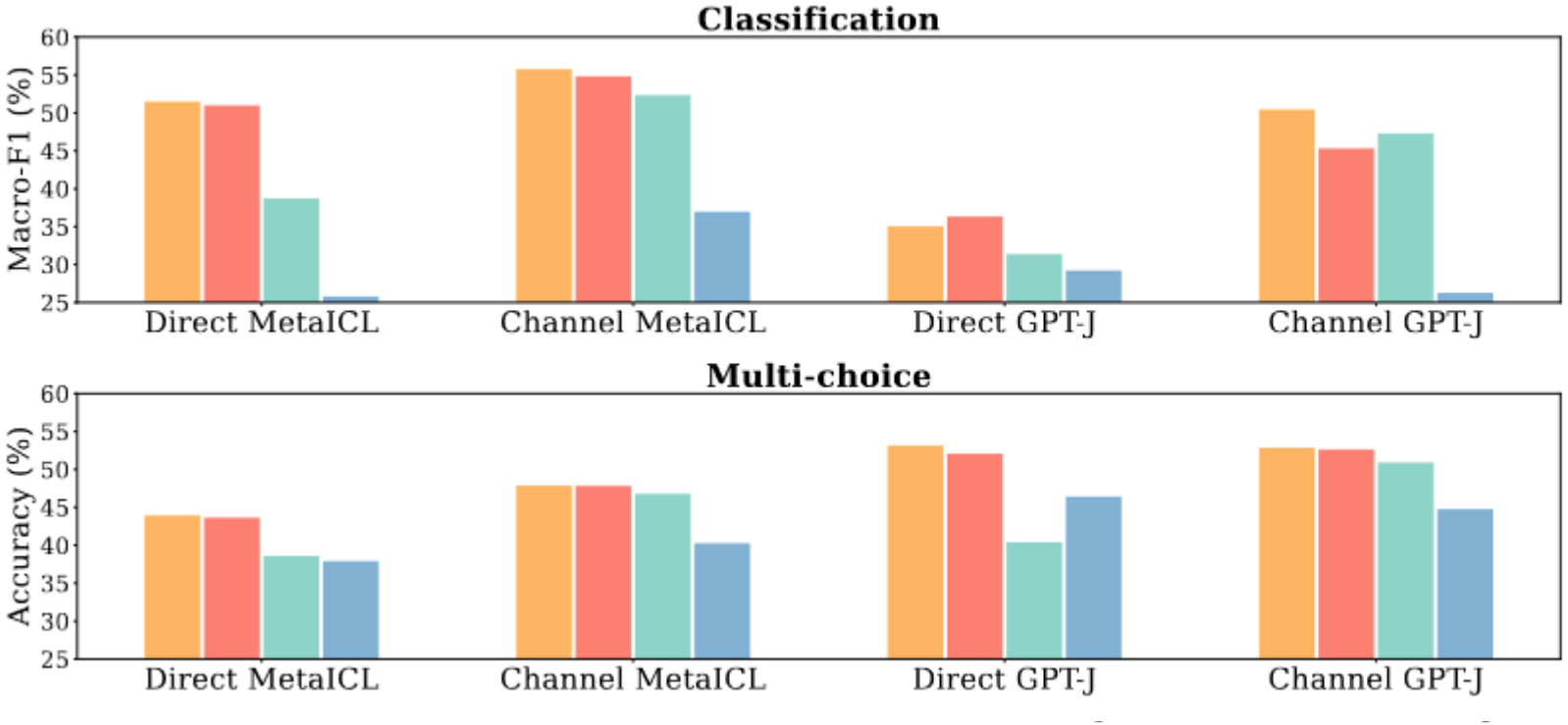


	F	*L*	*I*	*M*
Gold labels	✓	✓	✓	✓
Random labels	✓	✓	✓	✗
OOD + Random labels	✓	✓	✗	✗
No demonstrations	✗	✗	✗	✗

F: Format
L: Label space
I: Input distribution
M: Input-Label Mapping

This suggests that in-distribution inputs in the demonstrations substantially contribute to performance gains.

Impact of the label space

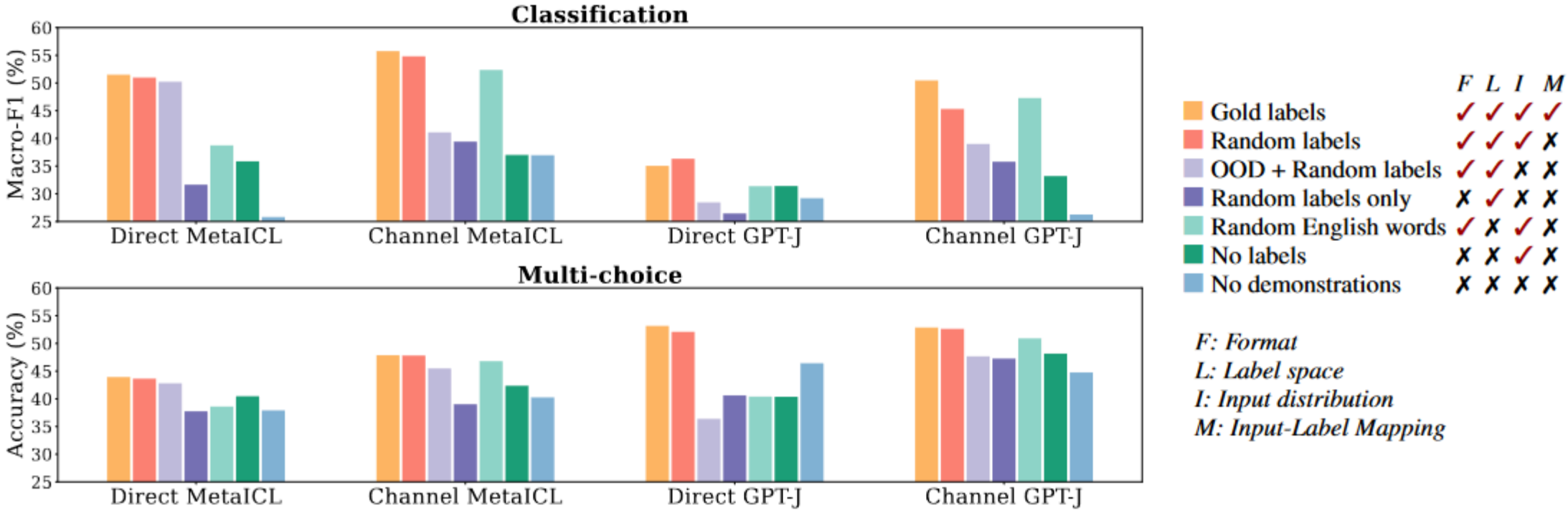


	F	*L*	*I*	*M*
Gold labels	✓	✓	✓	✓
Random labels	✓	✓	✓	✗
Random English words	✓	✗	✓	✗
No demonstrations	✗	✗	✗	✗

F: Format
L: Label space
I: Input distribution
M: Input-Label Mapping

This indicates that conditioning on the label space significantly contributes to performance gains.

Impact of input-label pairing



More interestingly, keeping the format plays a significant role in retaining a large portion of performance gains by only using the inputs or only using the labels.

Keeping the format of the input-label pairs is key.

Discussion & Conclusion

- **Does the model learn at test time?**

If we take a strict definition of learning: capturing the input-label correspondence given in the training data, then our findings suggest that LMs do not learn new tasks at test time. Our analysis shows that the model may ignore the task defined by the demonstrations and instead use prior from pretraining.

However, learning a new task can be interpreted more broadly: it may include adapting to specific input and label distributions and the format suggested by the demonstrations, and ultimately getting to make a prediction more accurately. With this definition of learning, the model does learn the task from the demonstrations.

Discussion & Conclusion

- **Capacity of LMs**

The model performs a downstream task without relying on the input-label correspondence from the demonstrations. This suggests that the model has learned the (implicit notion of) input-label correspondence from the language modeling objective alone, e.g., associating a positive review with the word ‘positive’.

Whether we need a better way of extracting the input-label mappings that are already stored in the LM, a better variant of the LM objective that learns a wider range of task semantics, or explicit supervision through fine-tuning on the labeled data?

Discussion & Conclusion

- **Connection to instruction-following models**

We think the demonstrations and instructions largely have the same role to LMs, and hypothesize that our findings hold for instruction-following models: **the instructions prompt the model to recover the capacity it already has, but do not supervise the model to learn novel task semantics**. This has been partially verified by Webson and Pavlick (2022) who showed that the model performance does not degrade much with irrelevant or misleading instructions.

Discussion & Conclusion

- **Significantly improved zero-shot performance**

One of our key findings is that it is possible to achieve nearly k-shot performance without using any labeled data, by simply pairing each unlabeled input with a random label and using it as the demonstrations. This means our zero-shot baseline level is significantly higher than previously thought. Future work can further improve the zero-shot performance with relaxed assumptions in access to the unlabeled training data.

LARGER LANGUAGE MODELS DO IN-CONTEXT LEARNING DIFFERENTLY

Jerry Wei[1,2,*] **Jason Wei**[1] **Yi Tay**[1] **Dustin Tran**[1] **Albert Webson**[1,3,*]

Yifeng Lu[1] **Xinyun Chen**[1] **Hanxiao Liu**[1] **Da Huang**[1] **Denny Zhou**[1]

Tengyu Ma[1,2,†]

[1] Google Research, Brain Team [2] Stanford University [3] Brown University

Motivation

- To successfully perform ICL, models can (a) mostly use semantic prior knowledge to predict labels while following the format of in-context exemplars and/or (b) learn the input–label mappings from the presented.
- Prior work on which of these factors drives performance is mixed. For instance, although Min et al. (2022b) showed that presenting random ground truth mappings in-context does not substantially affect performance (suggesting that models primarily rely on semantic prior knowledge), other work has shown that transformers in simple settings (without language modeling pretraining) implement learning algorithms such as ridge regression and gradient descent (Akyürek et al., 2023; von Oswald et al., 2022; Dai et al., 2022).

Introduction

In this paper, we study how these two factors—semantic priors and input–label mappings—interact in several experimental settings :

- In regular ICL, both semantic priors and input–label mappings can allow the model to perform in-context learning successfully.
- In flipped-label ICL, all labels in the exemplars are flipped, which means that semantic prior knowledge and input–label mappings disagree. Labels for the evaluation set stay the same, so for binary classification tasks, performing better than 50% accuracy in this setting means that the model is unable to override semantic priors, and performing below 50% accuracy means that the model is able to learn input–label mappings and override semantic priors.
- In semantically-unrelated label ICL (SUL-ICL), the labels are semantically unrelated to the task (e.g., for sentiment analysis, we use “foo/bar” instead of “negative/positive”). Since the semantic priors from labels are removed, the model can only perform ICL by using input–label mappings.

Introduction

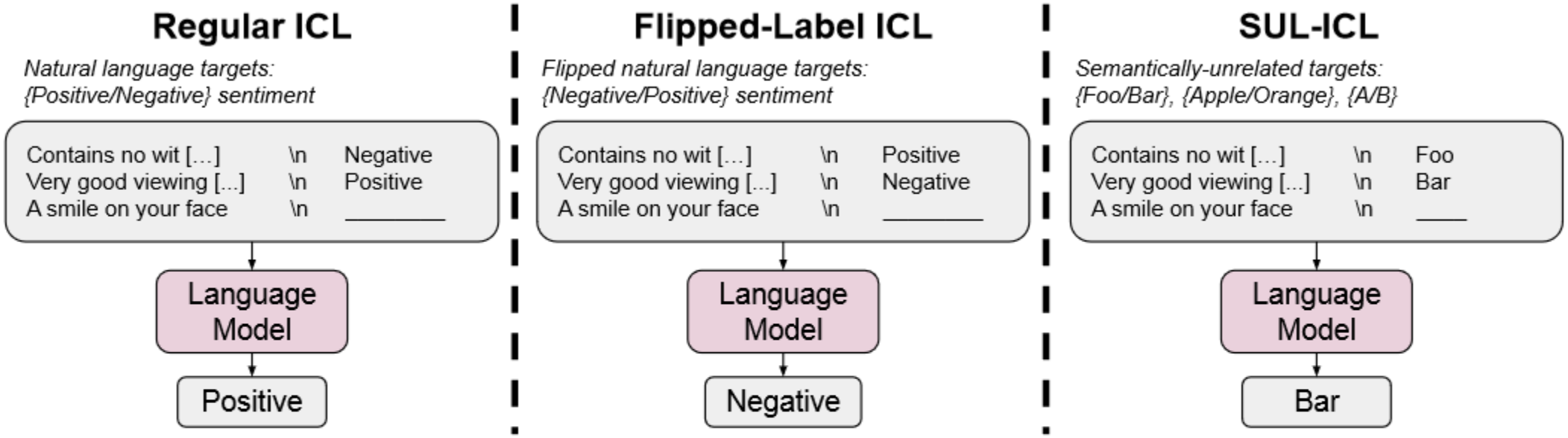


Figure 1: An overview of flipped-label ICL and semantically-unrelated label ICL (SUL-ICL), compared with regular ICL. Flipped-label ICL uses flipped targets, forcing the model override semantic priors in order to follow the in-context exemplars. SUL-ICL uses targets that are not semantically related to the task, which means that models must learn input–label mappings in order to perform the task because they can no longer rely on the semantics of natural language targets.

Experimental Setup

We run experiments in these settings spanning multiple model families with varying sizes, training data, and instruction tuning (GPT-3, InstructGPT, Codex, PaLM, Flan-PaLM) in order to analyze the interplay between semantic priors and input–label mappings, paying special attention to how results change with respect to model scale.

We experiment on seven NLP tasks that have been widely used in the literature. The seven tasks are: Sentiment Analysis; Subjective/Objective Sentence Classification; Question Classification; Duplicated-Question Recognition; Textual Entailment Recognition; Financial Sentiment Analysis; and Hate Speech Detection.

Experimental Setup

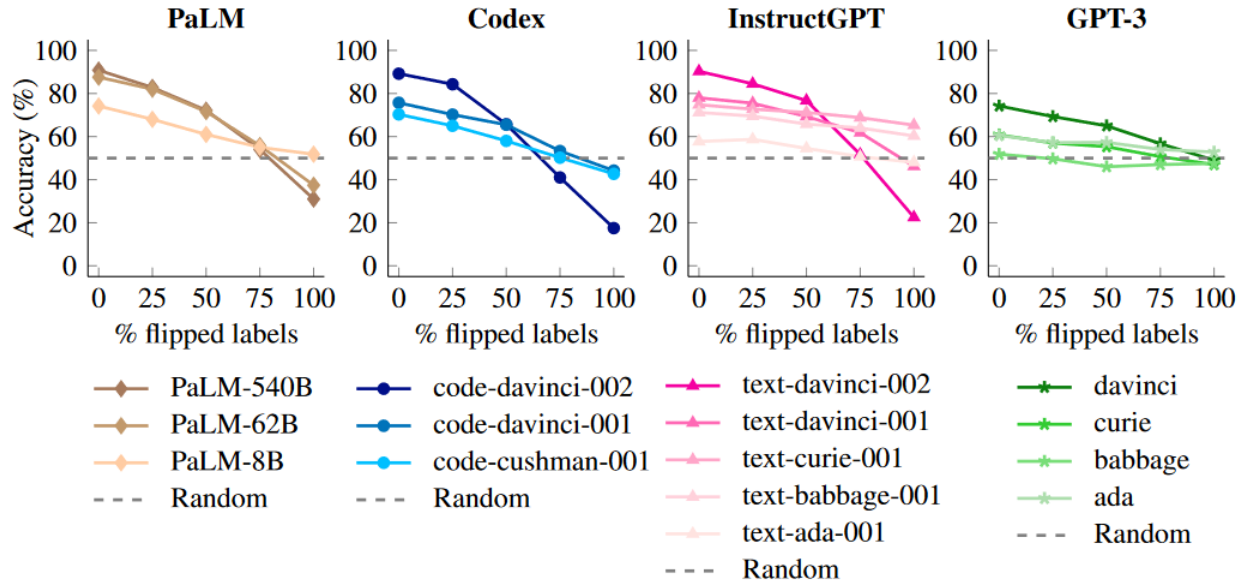
Model Family	Model Name (Abbreviation)
GPT-3	ada (a), babbage (b), curie (c), davinci (d)
InstructGPT	text-ada-001 (a-1), text-babbage-001 (b-1), text-curie-001 (c-1), text-davinci-001 (d-1), text-davinci-002 (d-2)
Codex	code-cushman-001 (c-c-1), code-davinci-001 (c-d-1), code-davinci-002 (c-d-2)
PaLM	PaLM-8B, PaLM-62B, PaLM-540B
Flan-PaLM	Flan-PaLM-8B, Flan-PaLM-62B, Flan-PaLM-540B

Table 1: Models used in this paper.

Experimental Setup

By default, we use k=16 in-context exemplars per class, though we also experiment with varying number of exemplars. We also use the “Input/Output” template for, with ablations for input format shown in Appendix, and the semantically-unrelated “Foo”/“Bar” targets as shown in Figure (ablations for target type are shown in Appendix). Finally, to reduce inference costs, we use 100 randomly sampled evaluation examples per dataset, as it is more beneficial to experiment with a more-diverse range of datasets and model families than it is to include more evaluation examples per dataset, and our research questions depend more on general behaviors than on small performance deltas.

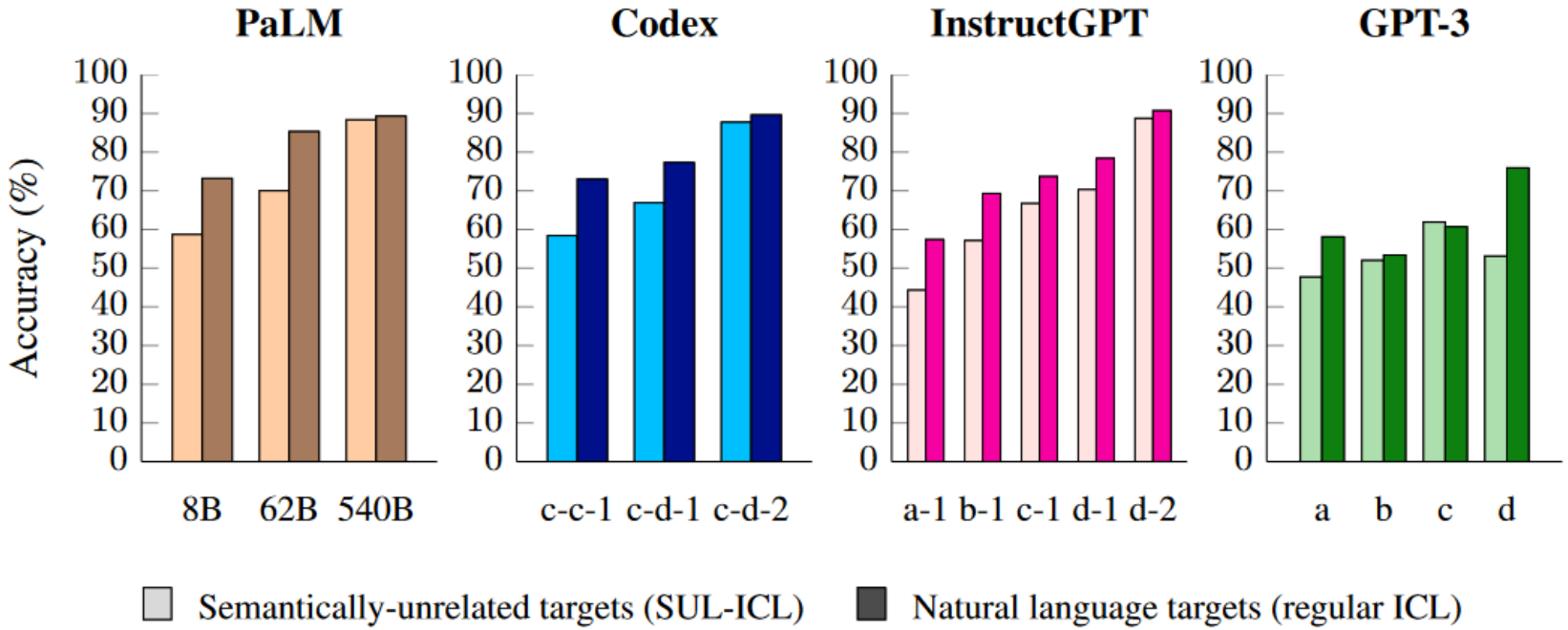
Input–label Mappings Override Semantic Priors In Large Models



These results indicate that large models can override prior knowledge from pretraining with input-label mappings presented in-context. Small models, on the other hand, do not flip their predictions and thus are unable to override semantic priors (consistent with Min et al. (2022b)). Because this ability to override prior knowledge with input–label mappings only appears in large models, we conclude that it is an emergent phenomena unlocked by model scaling.

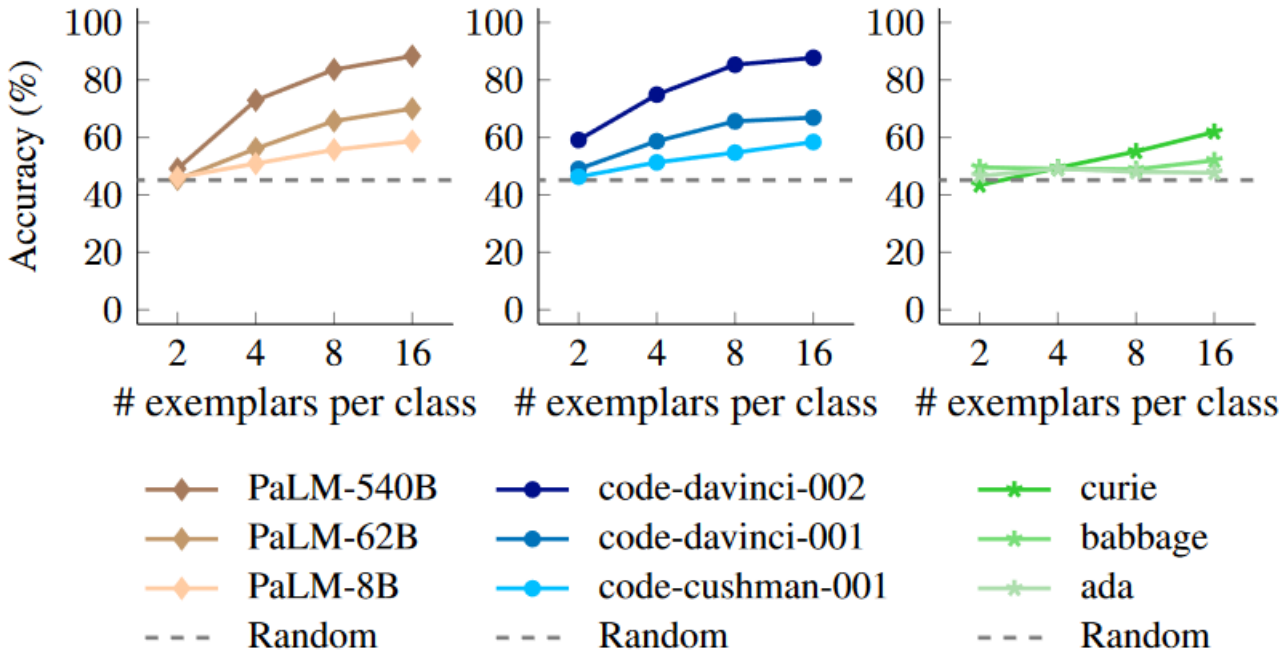
Note that GPT-3 models can remove semantic priors but cannot override them, even when presented with 100% flipped labels. For this reason, we consider all GPT-3 models to be "small" models because they all behave similarly to each other this way.

In-context Learning With Semantically Unrelated Labels Emerges With Scale



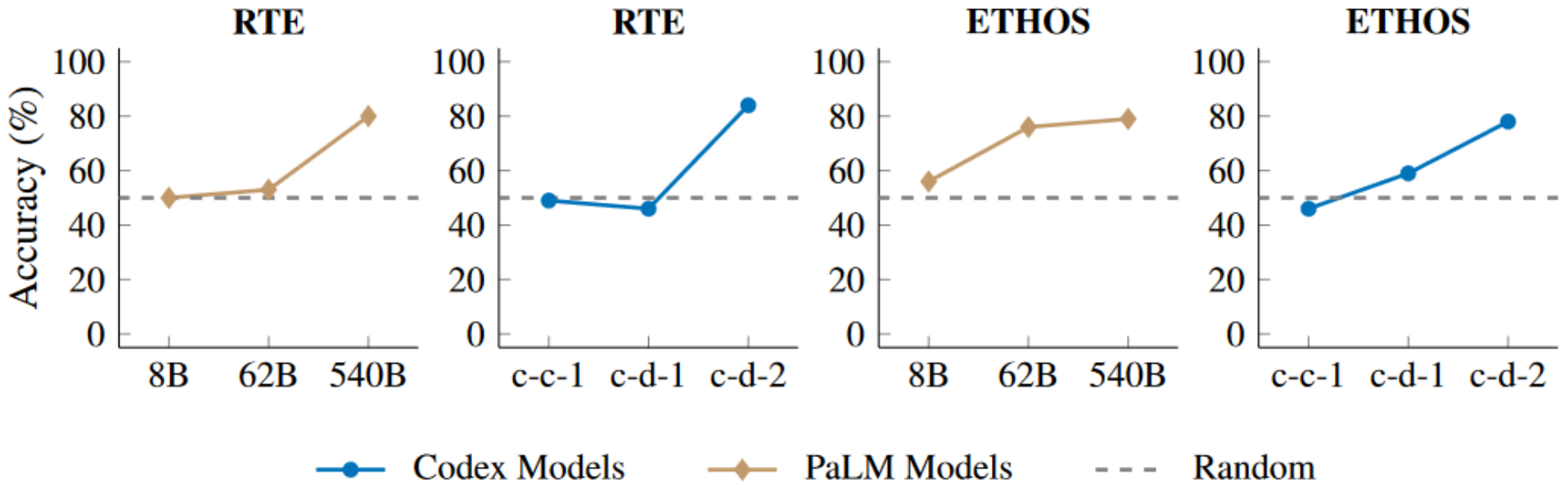
Because small models are heavily affected when the semantic meaning of targets is removed, we conclude that they primarily rely on the semantic meaning of targets for in-context learning rather than learn the presented input–label mappings. Large models, on the other hand, experience very small performance drops after this change, indicating that they have the ability to learn input–label mappings in-context when the semantic nature of targets is removed. Hence, the ability to learn input–label mappings in-context without being given semantic priors can also be seen as an emergent ability of model scale.

In-context Learning With Semantically Unrelated Labels Emerges With Scale



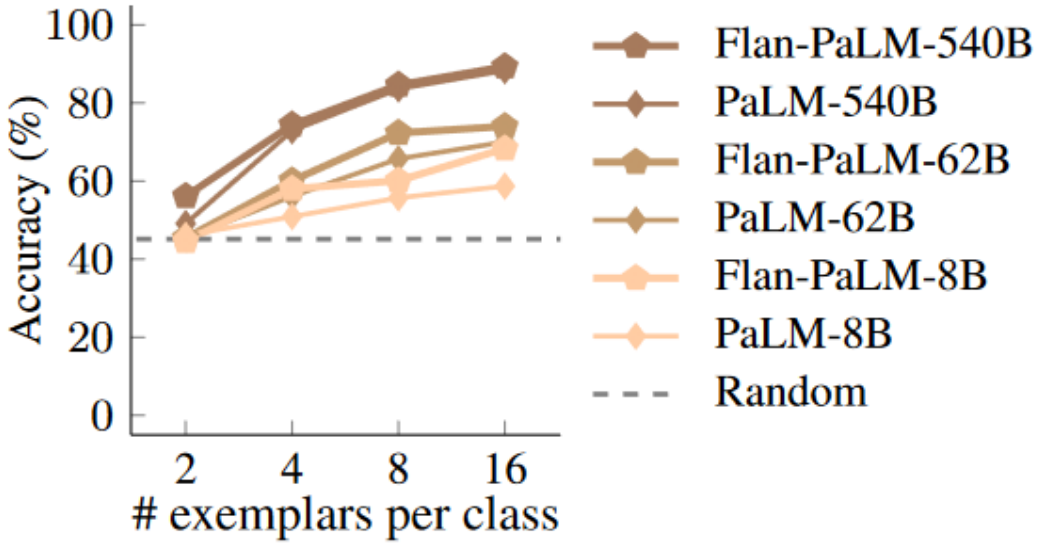
We find that for the three model families that we tested, including more in-context exemplars results in a greater performance improvement for large models than it does for small models. This indicates that large models are better at learning from in-context exemplars than small models are, implying that large models are more capable of using the additional input–label mappings presented in context to better learn the correct relationships between inputs and labels.

In-context Learning With Semantically Unrelated Labels Emerges With Scale



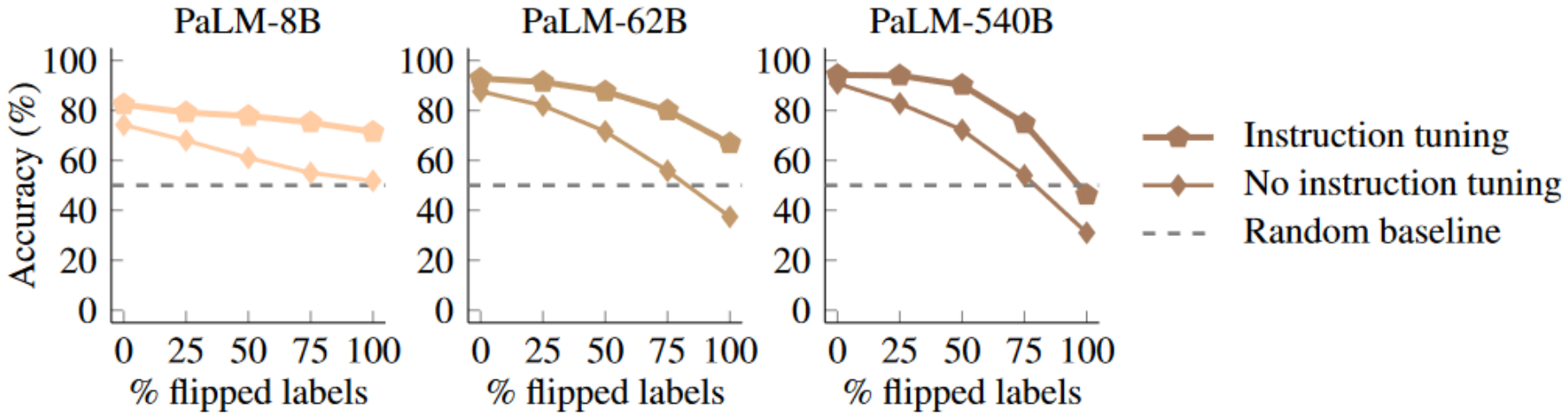
For many datasets that do not show emergence, even small models can outperform random guessing without many in-context exemplars (e.g., on SST-2, TREC, SUBJ, FP). These results show another example of how, for some tasks, the ability to learn input–label mappings in-context without being given semantic priors is only emergent in large-enough language models.

Instruction Tuning With Exemplars Improves Input–Label Mappings Learning And Strengthens Semantic Priors



This trend suggests that instruction tuning strengthens the ability to learn input–label mappings (an expected outcome).

Instruction Tuning With Exemplars Improves Input–Label Mappings Learning And Strengthens Semantic Priors



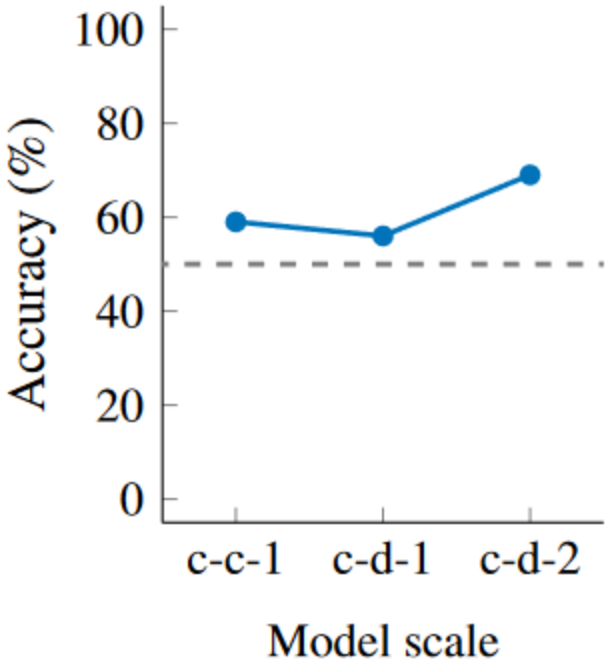
These results indicate that instruction tuning either increases the extent to which models rely on semantic priors when they are available or gives models more semantic priors, as instruction-tuned models are less capable of flipping their natural language targets to follow the flipped labels that were presented. Combined with the result from Figure 6, we conclude that although instruction tuning improves the ability to learn input–label mappings, it concurrently strengthens the usage of semantic priors, similar to the findings in Min et al. (2022a).

Large Language Models Can Perform Linear Classification

Algorithm 1 Generating one evaluation example for N-dimensional linear classification ($y = a_1x_1 + ... + a_Nx_N$) with k in-context exemplars per class. Random N-D vectors are generated using `np.random.randint()`.

procedure GENERATEEVAL(N, k)
  $a \leftarrow$ random N-D vector ▷ Ground-truth coefficients
  $p \leftarrow$ random N-D vector ▷ A pivot point
  $t = \langle a, p \rangle$ ▷ Threshold between positive and negative examples
  $x_{train} \leftarrow [\,], y_{train} \leftarrow [\,]$
  for $i \leftarrow 1$ to k **do** ▷ $2k$ in-context exemplars
    $x_+ \leftarrow$ random N-D vector conditioned on $\langle x_+, a \rangle > t$ ▷ Positive example
    $x_- \leftarrow$ random N-D vector conditioned on $\langle x_-, a \rangle \leq t$ ▷ Negative example
    $x_{train} \leftarrow x_{train} + [x_+, x_-]$
    $y_{train} \leftarrow y_{train} + [1, -1]$
  end for
  $x_{eval} \leftarrow$ random N-D vector
  $y_{eval} \leftarrow 1$ **if** $\langle x_{eval}, a \rangle > t$, **else** -1
  return $x_{train}, y_{train}, x_{eval}, y_{eval}$
end procedure

Large Language Models Can Perform Linear Classification



In Figure 8, we show Codex model performance on N=16 dimensional linear classification. We find that the largest model outperforms random guessing by 19% on this task, while smaller models cannot outperform random guessing by more than 9%. These results suggest that there exists some scaling factor that allows large-enough language models to perform high-dimensional linear classification.