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

Anonymous ACL submission

Abstract

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. In this paper, we show that ground truth demonstrations are in fact not required—randomly replacing labels in the demonstrations barely hurts performance, consistently over 12 different models including GPT-3. Instead, we find that other aspects of the demonstrations are the key drivers of end task performance, including the fact that they provide a few examples of (1) the label space, (2) the distribution of the input text, and (3) the overall format of the sequence. Together, our analysis provides a new way of understanding how and why in-context learning works, while opening up new questions about how much can be learned from large language models through inference alone.

1 Introduction

Large language models (LMs) have shown impressive performance on downstream tasks by simply conditioning on a few input-label pairs (demonstrations); this type of inference has been referred to as *in-context learning* (Brown et al., 2020). Despite in-context learning consistently outperforming zero-shot inference on a wide range of tasks (Zhao et al., 2021; Liu et al., 2021), there is little understanding of *how* it works and *which* aspects of the demonstrations contribute to end task performance.

In this paper, we show that ground truth demonstrations are in fact not required for effective in-context learning (Section 4). Specifically, replacing the labels in demonstrations with random labels barely hurts performance (Figure 1). The result is consistent over 12 different models including the GPT-3 family (Radford et al., 2019; Min et al., 2021b; Wang and Komatsuzaki, 2021; Artetxe et al., 2021; Brown et al., 2020). This strongly suggests, counter-intuitively, that the model *does not* rely on the input-label mapping in the demonstrations to perform the task.

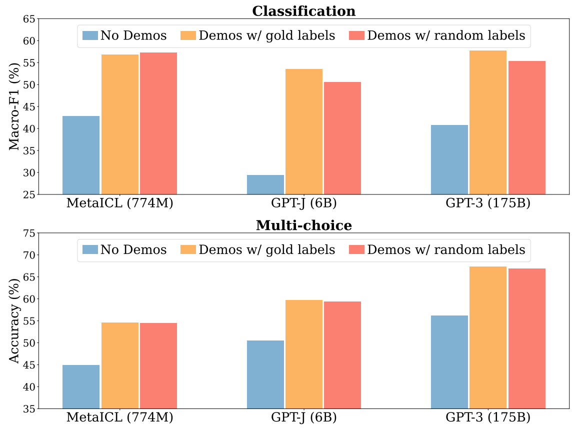

Figure 1: Results in classification (top) and multi-choice tasks (bottom), using three LMs with varying size. Reported on six datasets on which GPT-3 is evaluated; the channel method is used. See Section 4 for the full results. In-context learning performance drops only marginally when labels in the demonstrations are replaced by random labels.

Further analysis investigates which parts of demonstrations actually *do* contribute to the performance. We identify possible aspects of demonstrations (e.g., the label space and the distribution of the input text) and evaluate a series of variants of the demonstrations to quantify the impact of each (Section 5). We find that: (1) the label space and the distribution of the input text *specified by* the demonstrations are both key to in-context learning (regardless of whether the labels are correct for individual inputs); (2) specifying the overall format is also crucial, e.g., when the label space is unknown, using random English words as labels is significantly better than using no labels; and (3) meta-training with an in-context learning objective (Min et al., 2021b) magnifies these effects—the models almost exclusively exploit simpler aspects of the demonstrations like the format rather than the input-label mapping.

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 (1) counter-intuitively does not rely on the ground truth input-label mapping provided in the demonstrations as much as we thought (Section 4), and (2) nonetheless still benefits from knowing the label space and the distribution of inputs specified by the demonstrations (Section 5). 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 (Section 6).

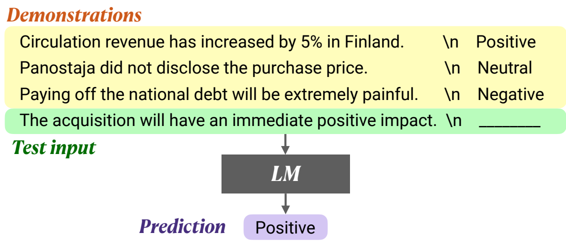

Figure 2: An overview of in-context learning. The demonstrations consist of k input-label pairs from the training data ($k = 3$ in the figure).

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‡	✗	✗

Table 1: A list of LMs used in the experiments: GPT-2 (Radford et al., 2019), MetaICL (Min et al., 2021b), GPT-J (Wang and Komatsuzaki, 2021), fairseq LMs (Artetxe et al., 2021) and GPT-3 (Brown et al., 2020). 'Public' indicates whether the model weights are public; 'Meta-trained' indicates whether the model is meta-trained with an in-context learning objective. †We use dense models in Artetxe et al. (2021) and refer them as fairseq LMs for convenience. ‡We use the Davinci API (the *base* version, not the *instruct* version) and assume it to be 175B, following Gao et al. (2021) and Artetxe et al. (2021).

2 Related Work

Large language models have been key to strong performance in a wide range of downstream tasks (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019; Raffel et al., 2020; Lewis et al., 2020). While finetuning has been a popular approach to transfer to new tasks (Devlin et al., 2019), it is often impractical to finetune a very large model (e.g. $\geq$10B parameters). Brown et al. (2020) propose in-context learning as an alternative way to learn a new task. As depicted in Figure 2, the LM learns a new task via inference alone by conditioning on a concatenation of the training data as demonstrations, without any gradient updates.

In-context learning has been the focus of significant study since its introduction. Prior work proposes better ways of formulating the problem (Zhao et al., 2021; Holtzman et al., 2021; Min et al., 2021a), better ways of choosing labeled examples for the demonstrations (Liu et al., 2021; Lu et al., 2021; Rubin et al., 2021), meta-training with an explicit in-context learning objective (Chen et al., 2021; Min et al., 2021b), and learning to follow instructions as a variant of in-context learning (Mishra et al., 2021b; Efrat and Levy, 2020; Wei et al., 2022; Sanh et al., 2022). At the same time, some work reports brittleness and over-sensitivity for in-context learning (Lu et al., 2021; Zhao et al., 2021; Mishra et al., 2021a).

Relatively less work has been done to understand why in-context learning works. Xie et al. (2022) provide theoretical analysis that in-context learning can be formalized as Bayesian inference that uses the demonstrations to recover latent concepts. Razeghi et al. (2022) show that in-context learning performance is highly correlated with term frequencies in the pretraining data. To the best of our knowledge, this paper is the first that provides an empirical analysis that investigates why in-context learning achieves performance gains over zero-shot inference. We find that the ground truth input-label mapping in the demonstrations has only a marginal effect, and measure the impact of finer-grained aspects of the demonstrations.

3 Experimental Setup

We describe the experimental setup used in our analysis (Section 4 and 5).

Models. We experiment with 12 models in total. We include 6 language models (Table 1), all of which are decoder-only, dense LMs. We use each LM with two inference methods, direct and channel, following Min et al. (2021a). The sizes of LMs vary from 774M to 175B. We include the largest dense LM (GPT-3) and the largest publicly released dense LM (fairseq 13B) at the time of con-

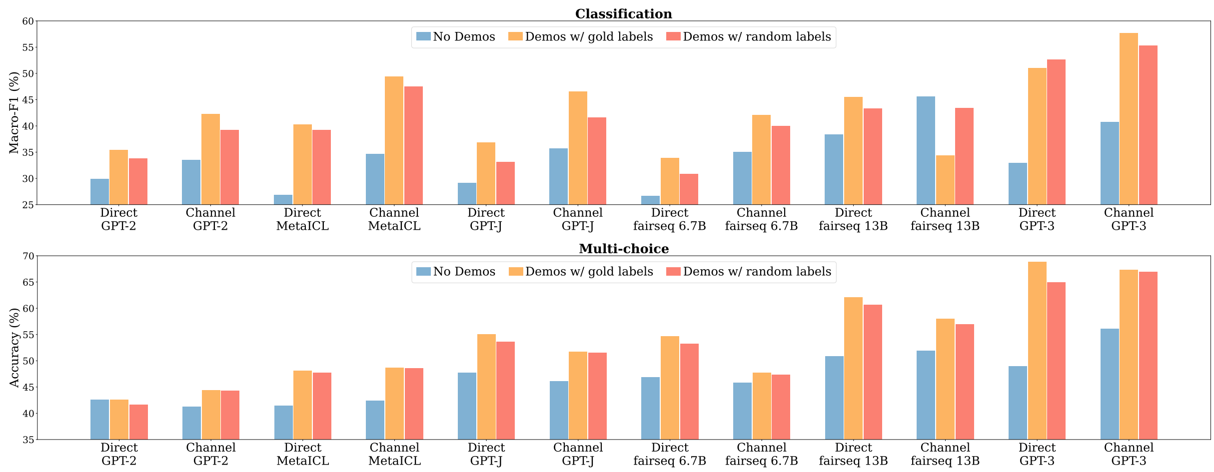

Figure 3: Results when using no-demonstrations, demonstrations with gold labels, and demonstrations with random labels in classification (top) and multi-choice tasks (bottom). Note that the first eight models are evaluated on 16 classification and 10 multi-choice datasets, and the last four models are evaluated on 3 classification and 3 multi-choice datasets. See Figure 11 for numbers comparable across all models. **Model performance with random labels is very close to performance with gold labels** (more discussion in Section 4.1).

ducting experiments. We also include MetaICL, which is initialized from GPT-2 Large and then meta-trained on a collection of supervised datasets with an in-context learning objective, and ensure that our evaluation datasets do not overlap with those used at meta-training time.

Evaluation Data. We evaluate on 26 datasets, including sentiment analysis, paraphrase detection, natural language inference, hate speech detection, question answering, and sentence completion (full list and references provided in Appendix A).[1] We use these datasets because they (1) are true low-resource datasets with less than 10K training examples, (2) include well-studied benchmarks from GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019a), and (3) cover diverse domains including science, social media, finance, and more. The 26 datasets can be further broken down into 16 classification tasks and 10 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. For fairseq 13B and GPT-3, due to limited resources, we experiment with a subset of 6 datasets[2] and 3 random seeds. We report Macro-F1 for classification tasks and Accuracy for multi-choice tasks. We compute per-dataset average over seeds, and then report macro-average over datasets. We use the minimal templates in forming an input sequence from an example. We refer to Appendix B for more details.

4 Ground Truth Matters Little

4.1 Gold labels vs. random labels

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.[3]

No demonstrations is a typical zero-shot method that does not use any labeled data. A prediction is made via $\text{argmax}_{y \in \mathcal{C}} P(y|x)$, where x is the test input and $\mathcal{C}$ 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 $\text{argmax}_{y \in \mathcal{C}} P(y|x_1, y_1...x_k, y_k, x)$.

Demonstrations w/ random labels are formed with random labels, instead of gold labels from the labeled data. Each x_i ($1 \leq i \leq k$) is paired with $\tilde{y}_i$ that is randomly sampled at uniform from $\mathcal{C}$. A concatenation of

[1] For convenience, we use 'labels' to refer to the output for the task, though our datasets include non-classification tasks.

[2] Three classification and three multi-choice: MRPC, RTE, Tweet_eval-hate, OpenbookQA, CommonsenseQA, COPA.

[3] Without loss of generality, all methods in Section 4 and 5 are described based on the direct method, but can be trivially converted to the channel method by flipping x and y.

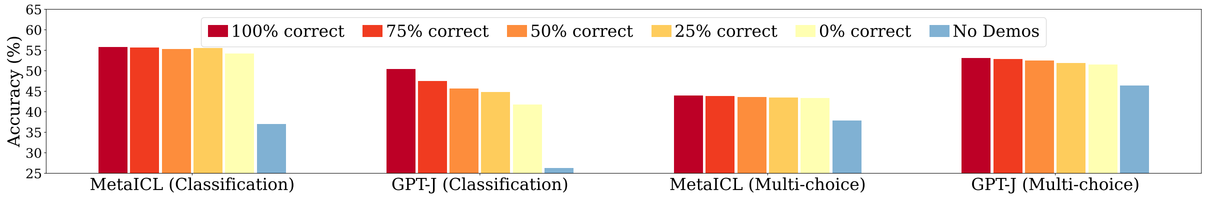

Figure 4: Results with varying number of correct labels in the demonstrations. Channel and Direct used for classification and multi-choice, respectively. Performance with no demonstrations (blue) is reported as a reference.

$(x_1, \tilde{y}_1)...(x_k, \tilde{y}_k)$ is then used to make a prediction via $\text{argmax}_{y \in \mathcal{C}} P(y|x_1, \tilde{y}_1...x_k, \tilde{y}_k, x)$.

Results are reported in Figure 3. First, using the demonstrations with gold labels significantly improves the performance over no demonstrations, as it has been consistently found in much of prior work (Brown et al., 2020; Zhao et al., 2021; Liu et al., 2021). We then find that **replacing gold labels with random labels only marginally hurts performance**. The trend is consistent over nearly all models: models see performance drop in the range of 0–5% absolute. There is less impact in replacing labels in multi-choice tasks (1.7% on average) than in classification tasks (2.6% absolute).

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.

It is also worth noting that there is particularly little performance drop in MetaICL: 0.1–0.9% absolute. This suggests that meta-training with an explicit in-context learning objective actually encourages the model to essentially ignore the input-label mapping and exploit other components of the demonstrations (more discussion in Section 5.4).

4.2 Ablations

For additional ablations, we experiment with 5 classification and 4 multi-choice datasets.[4]

[4]Classification includes: MRPC, RTE, Tweet_eval-hate, SICK, poem-sentiment; Multi-choice includes OpenbookQA, CommonsenseQA, COPA and ARC.

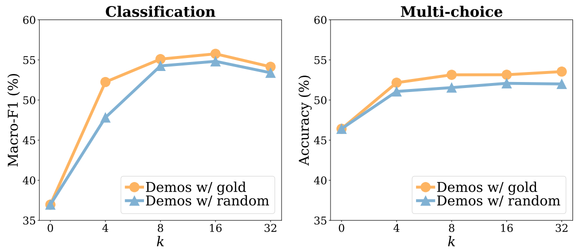

Figure 5: Ablations on varying numbers of examples in the demonstrations (k). Models that are the best under 13B in each task category (Channel MetaICL and Direct GPT-J, respectively) are used.

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. We evaluate "Demonstrations w/ a% correct labels" ($0 \leq a \leq 100$) which consist of $k \times a/100$ correct pairs and $k \times (1 - a/100)$ incorrect pairs (see Algorithm 1 in Appendix B). Here, $a = 100$ is the same as typical in-context learning, i.e., demonstrations w/ gold labels.

Results are reported in Figure 4. Model performance is fairly insensitive to the number of correct labels in the demonstrations. In fact, always using incorrect labels significantly outperforms no-demonstrations, e.g., preserving 92%, 100% and 97% of improvements from using the demonstrations with MetaICL in classification, MetaICL in multi-choice, and GPT-J in multi-choice, respectively. GPT-J in classification is an outlier where performance depends relatively more on the number of correct labels of the demonstrations—it achieves higher performance with a larger number of correct labels. Still, always using incorrect labels is significantly better than no demonstrations.

Is the result consistent with varying k? We study the impact of the number of input-label pairs (k) in the demonstrations. Results are reported in Figure 5. First, using the demonstrations significantly outperforms the no demonstrations method

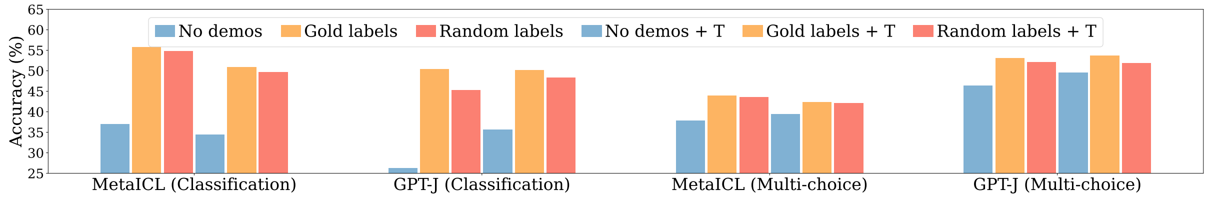

Figure 6: Results with minimal templates and manual templates. '+T' indicates that manual templates are used. Channel and Direct used for classification and multi-choice, respectively.

even with small k ($k = 4$), and performance drop from using gold labels to using random labels is consistently small across varying k, in the range of 0.8–1.6%.[5] Interestingly, model performance does not increase much as k increases when $k \geq 8$, both with gold labels and with random labels. This is in contrast with typical supervised training where model performance rapidly increases as k increases, especially when k is small. 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 (more discussion in Section 5).

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 (details in Appendix B). Figure 6 shows that the trend—replacing gold labels with random labels barely hurting performance—holds with manual templates. It is worth noting that using manual templates does not always outperform using minimal templates.

5 Why *does* In-Context Learning work?

Section 4 shows that the ground truth input-label mapping in the demonstrations has little impact to performance gains from in-context learning. This section further examines what other aspects of the demonstrations lead to good performance of in-context learning.

We identify four aspects of the demonstrations $(x_1, y_1)...(x_k, y_k)$ that potentially provide learning signal (depicted in Figure 7).

1. **The input-label mapping**, i.e., whether each input x_i is paired with a correct label y_i.

[5] With an exception of 4.4% in classification with $k = 4$, likely due to a high variance with a very small value of k.

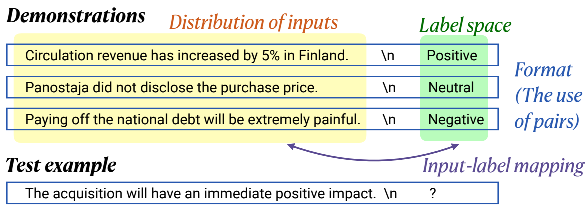

Figure 7: Four different aspects in the demonstrations: the input-label mapping, the distribution of the input text, the label space, and the use of input-label pairing as the format of the demonstrations.

2. **The distribution of the input text**, i.e., the underlying distribution that $x_1...x_k$ are from.
3. **The label space**, i.e., the space covered by $y_1...y_k$.
4. **The format**—specifically, the use of input-label pairing as the format.

As Section 4 does for the input-label mapping, we design a series of variants of the demonstrations that quantify the impact of each aspect in isolation (Section 5.1–5.3). We then additionally discuss the trend of the models meta-trained with an in-context learning objective (Section 5.4). For all experiments, models are evaluated on five classification and four multi-choice datasets as in Section 4.2. See Appendix B and Table 4 for implementation details and example demonstrations, respectively.

5.1 Impact of the distribution of the input text

We experiment with **OOD demonstrations** which include out-of-distribution (OOD) text instead of the inputs from unlabeled training data. Specifically, a set of k sentences $\{x_{i,\text{rand}}\}_{i=1}^{k}$ are randomly sampled from an external corpus, and replace $x_1...x_k$ in the demonstrations. This variant assesses the impact of the distribution of the input text, while keeping the label space and the format of the demonstrations.

Results. Figure 8 shows that using out-of-distribution inputs instead of the inputs from the

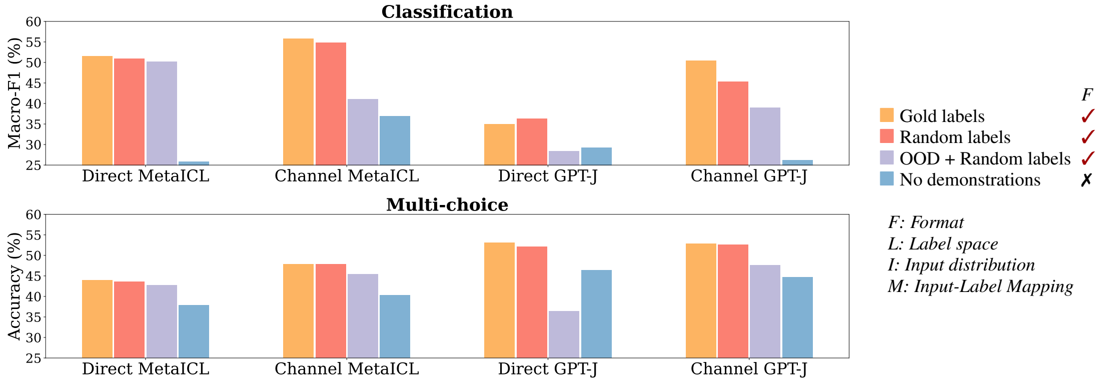

Figure 8: Impact of the distribution of the inputs. Evaluated in classification (top) and multi-choice (bottom). The impact of the distribution of the input text can be measured by comparing and . The gap is substantial, with an exception in Direct MetaICL (discussion in Section 5.1).

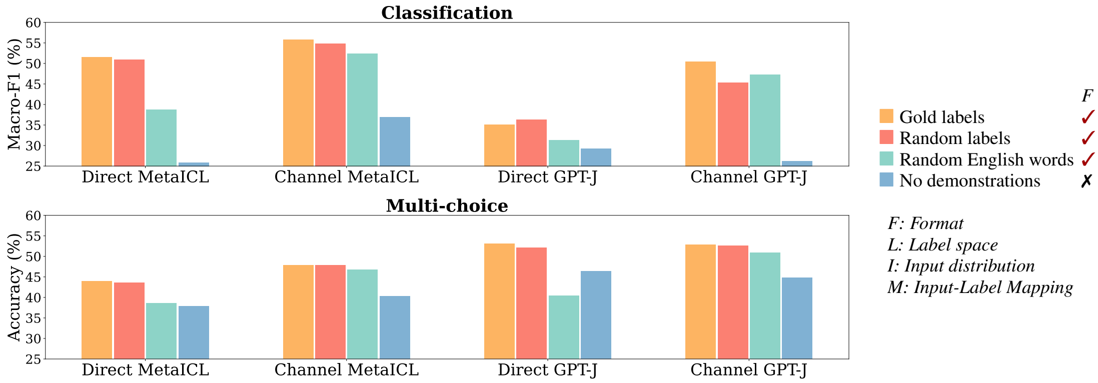

Figure 9: Impact of the label space. Evaluated in classification (top) and multi-choice (bottom). The impact of the label space can be measured by comparing and . The gap is significant in the direct models but not in the channel models (discussion in Section 5.2).

training data significantly drops the performance when Channel MetaICL, Direct GPT-J or Channel GPT-J are used, both in classification and multi-choice, by 3–16% in absolute. In the case of Direct GPT-J in multi-choice, it is even significantly worse than no demonstrations. Direct MetaICL is an exception, which we think is the effect of meta-training (discussion in Section 5.4).

This suggests that in-distribution inputs in the demonstrations substantially contribute to performance gains. This is likely because conditioning on the in-distribution text makes the task closer to language modeling, since the LM always conditioned on the in-distribution text during training.

5.2 Impact of the label space

We also experiment with **demonstrations w/ random English words** that use random English words as labels for all k pairs. Specifically, we sample a random subset of English words $\mathcal{C}_{\text{rand}}$ where $|\mathcal{C}_{\text{rand}}| = |\mathcal{C}|$, and randomly pair $\tilde{y}_i \in \mathcal{C}_{\text{rand}}$ with x_i. This variant assesses the impact of the label space, while keeping the distribution of the input text and the format of the demonstrations.

Results. Based on Figure 9, direct models and channel models exhibit different patterns. With direct models, the performance gap between using random labels within the label space and using random English words is significant, ranging between 5–16% absolute. This indicates that conditioning on the label space significantly contributes to performance gains. This is true even for multi-choice tasks where there is no fixed set of labels—we hypothesize that multi-choice tasks still do have a particular distribution of the choices (e.g., objects like “Bolts” or “Screws” in the OpenBookQA dataset) that the model uses.

On the other hand, removing the output space does not lead to significant drop in the channel models: there is 0–2% drop in absolute, or sometimes even an increase. We hypothesize that this is because the channel models only condition on the

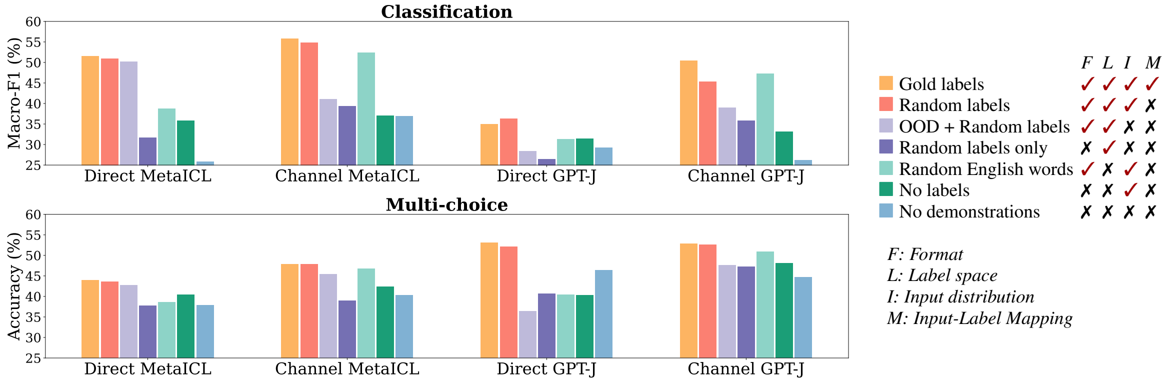

Figure 10: Impact of the format, i.e., the use of the input-label pairs. Evaluated in classification (top) and multi-choice (bottom). Variants of demonstrations without keeping the format (■ and ■) are overall not better than no demonstrations (■). Keeping the format is especially significant when it is possible to achieve substantial gains with the label space but without the inputs (■ vs. ■ in Direct MetaICL), or with the input distribution but without the labels (■ vs. ■ in Channel MetaICL and Channel GPT-J). More discussion in Section 5.3.

labels, and thus are not benefiting from knowing the label space. This is in contrast to direct models which must *generate* the correct labels.

5.3 Impact of input-label pairing

Section 5.1 and 5.2 focus on variants which keep the format of the demonstrations as much as possible. This section explores variants that change the format. While there are many aspects of the format, we make minimal modifications to remove the pairings of inputs to outputs. Specifically, we evaluate (1) **demonstrations with no labels** where the LM is conditioned on the concatenation of $x_1...x_k$, and (2) **demonstrations with labels only** where the LM is conditioned on the concatenation of $y_1...y_k$. These ablations provide the no-format counterparts of the 'demonstrations with random English words' and 'demonstrations with OOD inputs', respectively.

Results. Based on Figure 10, removing the format is close to or worse than no demonstrations, indicating the importance of the format. This is likely because conditioning on a sequence of input-label pairs triggers the model to mimic the overall format and complete the new example as expected when the test input is given.

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. For instance, with Direct MetaICL, it is possible to retain 95% and 82% of improvements from in-context learning (demonstrations with gold labels) by simply sampling random sentences from a corpus and randomly pairing them with the label set (■ in Figure 10) in classification and multi-choice, respectively. Similarly, with the channel models, it is possible to retain 82%, 87%, 86% and 75% of improvements from in-context learning by simply pairing each input from the unlabeled training data with a random English word (■ in Figure 10) in MetaICL classification, GPT-J classification, MetaICL multi-choice and GPT-J multi-choice, respectively. For all of these cases, removing inputs instead of using OOD inputs, or removing labels instead of using random English words is significantly worse, indicating that **keeping the format of the input-label pairs is key**.

5.4 Impact of meta-training

Different from other models, MetaICL is trained with an in-context learning objective, in line with recent work that uses multi-task training on a large collection of supervised datasets (called meta-training) for generalization to new tasks (Aghajanyan et al., 2021; Khashabi et al., 2020; Wei et al., 2022; Sanh et al., 2022). We aim to better understand the role of this meta-training in relation with our findings by closely examining the result of MetaICL. In particular, we observe that the patterns we see so far are significantly more evident with MetaICL than with other models. For instance, the ground truth input-label mapping matters even less, and keeping the format of the demonstrations matters even more. There is nearly zero influence of the input-label mapping and the input distribution in Direct MetaICL, and the input-label mapping and the output space in Channel MetaICL.

Based on this observation, we hypothesize that

meta-training encourages the model to exclusively exploit simpler aspects of the demonstrations and to ignore others. This is based on our intuition that (1) the input-label mapping is likely harder to exploit, (2) the format is likely easier to exploit, and (3) the space of the text that the model is trained to generate is likely easier to exploit than the space of the text that the model conditions on.[6]

6 Discussion & Conclusion

In this paper, we study the role of the demonstrations with respect to the success of in-context learning.[7] We find that the ground truth input-label mapping in the demonstrations matters significantly less than one might think—replacing gold labels with random labels in the demonstrations only marginally lowers the performance. We then identify a series of aspects in the demonstrations and examine which aspect actually contributes to performance gains. Results reveal that (1) gains are mainly coming from *independent* specification of the input space and the label space, (2) the models can still retain up to 95% of performance gains by using either the inputs only or the label set only if the right format is used, and (3) meta-training with an in-context learning objective magnifies these trends. Together, our findings lead to a set of broader indications about in-context learning, as well as avenues for future work.

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 experiments in Section 4.2 show that when the task is defined by the demonstrations to predict 'negative' to a positive review and 'positive' to a negative review, the model still predicts 'positive' and 'negative' to positive and negative reviews, respectively.

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. Our experiments indicate that the model *does* make use of aspects of the demonstrations and achieve performance gains.

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'. On one hand, this suggests that the language modeling objective has led to great zero-shot *capacity*, even if it is not always evident from the naive zero-shot *accuracy*. On the other hand, this suggests that in-context learning is unlikely to work on a task whose input-label correspondence is not already captured in the LM, e.g., when the task semantics are not close enough to language modeling.

Connection to instruction-following models. Prior work has found it promising to train the model that reads the natural language description of the task (called instructions) and performs a new task at inference (Mishra et al., 2021b; Efrat and Levy, 2020; Wei et al., 2022; Sanh et al., 2022). 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. We leave analysis on instruction-following models for future work.

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.[8] Gains from the demonstrations with random labels over the previous zero-shot method (no demonstrations) are up to 20% absolute in classification and up to 15% absolute in multi-choice tasks. Future work can further improve the zero-shot performance with relaxed assumptions in access to the unlabeled training data.

[6] That is, the direct model exploits the label space better than the input distribution, and the channel model exploits the input distribution better than the label space.

[7] We focus on the tasks from established NLP benchmarks that have *real* natural language inputs. Synthetic tasks with more limited inputs may actually use the labels more, as observed by Rong (2021).

[8] We take the perspective that using the unlabeled training data is permitted (Kodirov et al., 2015; Wang et al., 2019b; Schick and Schütze, 2021).

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. 2021. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of EMNLP*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

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 *NeurIPS*.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

T. Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *ArXiv*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

L Gao, S Biderman, S Black, L Golding, T Hoppe, C Foster, J Phang, H He, A Thite, N Nabeshima, et al. 2021. The pile: an 800gb dataset of diverse text for language modeling 2020. *arXiv preprint arXiv:2101.00027*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *The First Joint Conference on Lexical and Computational Semantics (SemEval)*.

Ari Holtzman, Peter West, Vered Schwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *EMNLP*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing format boundaries with a single qa system. In *Findings of EMNLP*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *AAAI*.

Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. 2015. Unsupervised domain adaptation for zero-shot learning. In *Proceedings of the IEEE international conference on computer vision*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In

Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *EMNLP: System Demonstrations*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Robert L Logan IV, Ivana Balaževic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *LREC*.

Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: Matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021a. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021b. MetaICL: Learning to learn in context. *arXiv preprint*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021b. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *ArXiv*.

Sebastian Nagel. 2016. CC-News. http://web.archive.org/save/http://commoncrawl.org/2016/10/news-dataset-available.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

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

Frieda Rong. 2021. Extrapolating to unnatural language processing with gpt-3's in-context learning: The good, the bad, and the mysterious. https://ai.stanford.edu/blog/in-context-learning.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han

Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *ICLR*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *NAACL-HLT*.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *TACL*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. Quarel: A dataset and models for answering questions about qualitative relationships. In *AAAI*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. In *EMNLP*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. 2019b. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *ICLR*.

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

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. In *EMNLP*.

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

A Full Datasets

We include 26 datasets as follows: financial_phrasebank (Malo et al., 2014), poem_sentiment (Sheng and Uthus, 2020), medical_questions_pairs (McCreery et al., 2020), glue-mrpc (Dolan and Brockett, 2005), glue-wnli (Levesque et al., 2012), climate_fever (Diggelmann et al., 2020), glue-rte (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), superglue-cb (de Marneffe et al., 2019), sick (Marelli et al., 2014) , hate_speech18 (de Gibert et al., 2018), ethos-national_origin (Mollas et al., 2020), ethos-race (Mollas et al., 2020), ethos-religion (Mollas et al., 2020), tweet_eval-hate (Barbieri et al., 2020), tweet_eval-stance_atheism (Barbieri et al., 2020), tweet_eval-stance_feminist (Barbieri et al., 2020), quarel (Tafjord et al., 2019a), openbookqa (Mihaylov et al., 2018), qasc (Khot et al., 2020), commonsense_qa (Talmor et al., 2019), ai2_arc (Clark et al., 2018), codah (Chen et al., 2019), superglue-copa (Gordon et al., 2012), dream (Sun et al., 2019), quartz-with_knowledge (Tafjord et al., 2019b), quartz-no_knowledge (Tafjord et al., 2019b). The choice of datasets is made following low-resource datasets in Min et al. (2021b), with the exact same set of k-shot train data using 5 random seeds. We use the HuggingFace version of the data (Lhoest et al., 2021) and use the development data for evaluation, following Ye et al. (2021). See Table 2 for statistics.

Dataset	# Train	# Eval
Task category: Sentiment analysis		
financial_phrasebank	1,811	453
poem_sentiment	892	105
Task category: Paraphrase detection		
medical_questions_pairs	2,438	610
glue-mrpc	3,668	408
Task category: Natural language inference		
glue-wnli	635	71
climate_fever	1,228	307
glue-rte	2,490	277
superglue-cb	250	56
sick	4,439	495
Task category: Hate speech detection		
hate_speech18	8,562	2,141
ethos-national_origin	346	87
ethos-race	346	87
ethos-religion	346	87
tweet_eval-hate	8,993	999
tweet_eval-stance_atheism	461	52
tweet_eval-stance_feminist	597	67
Task category: Question answering		
quarel	1,941	278
openbookqa	4,957	500
qasc	8,134	926
commonsense_qa	9,741	1,221
ai2_arc	1,119	299
Task category: Sentence completion		
codah	1665	556
superglue-copa	400	100
dream	6116	2040
quartz-with_knowledge	2696	384
quartz-no_knowledge	2696	384

Table 2: 26 datasets used for experiments, classified into 6 task categories. # Train and # Test indicate the number of training and test examples of the dataset. Note that # train is based on the original training dataset but we use k random samples for k-shot evaluation.

B Experimental Details

Example template We follow Ye et al. (2021); Min et al. (2021b); Logan IV et al. (2021) in using the minimal format to transform the input to a sequence (e.g. a concatenation of multiple inputs) and using the label words from each dataset as it is. We also explore manual templates taken from prior work (Holtzman et al., 2021; Zhao et al., 2021) as reported in Section 4.2, although we find that using these templates is not consistently better than using minimal templates. We thus run main experiments with minimal templates. Example templates are provided in Table 3.

Format of the demonstrations We follow the standard of each model for formatting the demonstrations, either from exploration in prior work or the example code provided in the official tutorial. For GPT-2, we separate the input and the label, and each demonstration example with a space. For MetaICL, GPT-J and GPT-3, we separate the input and the label with a newline (`\n`), and each demonstration example with three newlines. For fairseq models, we use a newline to separate the input and the label as well as each demonstration example.

Details in variants of the demonstrations For "demonstrations w/ a% accurate labels" ($0 \leq a \leq 100$), we use $k \times a/100$ correct pairs and $k \times (1 - a/100)$ incorrect pairs in a random order, as described in Algorithm 1. For "OOD demonstrations", we use CC-News (Nagel, 2016) as an external corpus. We consider the length of the text during sampling, so that sampled sentences have similar length to the test input. For "demonstrations with random English words", we use `pypi.org/project/english-words` for the set of En-

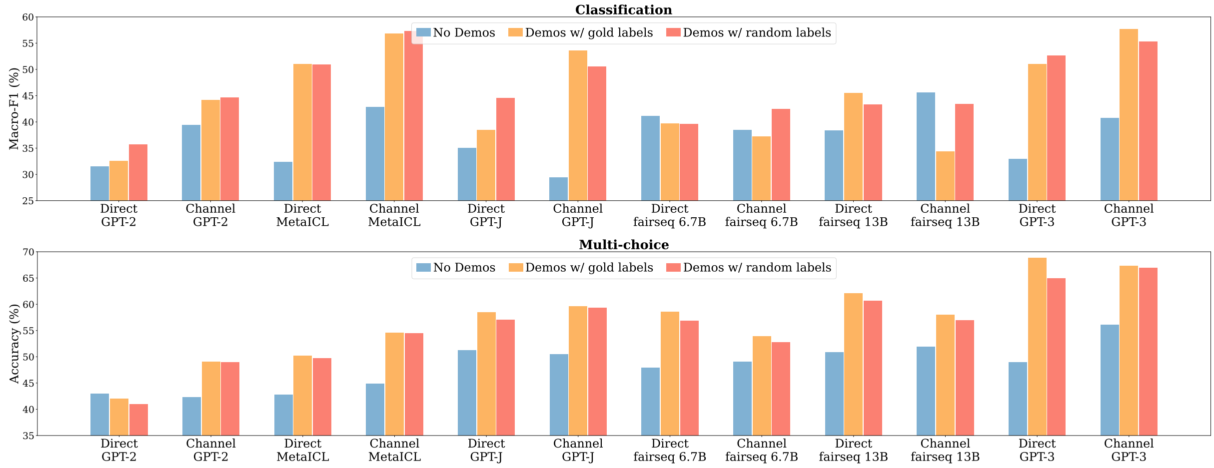

Figure 11: Results of No-demonstration, Gold demonstration and Random demonstration on 3 classification datasets (top) and 3 multi-choice datasets (bottom). Details in Section 4.1. This figure is for providing numbers that are comparable across models—full results with more datasets are reported in Figure 3.

Algorithm 1 Forming the demonstrations with an accuracy of $a\%$.

```
procedure FormDemons({(x_i, y_i)}_{i=1}^k, a)
    D ← []  // demonstration to be formed
    n ← k × a/100  // number of correct pairs
    G ← Sample(Range(1, k), n)
    for i ∈ G Range(1, k) do
        if i ∈ G then  // add correct pair
            D.append((x_i, y_i))
        else  // add incorrect pair
            D.append((x_i, Sample(C − y_i)))
    return D
```

glish words, which consists of 61,569 words.

Table 4 provides a list of example demonstrations for each method used in Section 5.

C More Experimental Results

C.1 Gold labels vs. random labels

Figure 11 shares the same interface as Figure 3, but all models are evaluated on 3 classification and 3 multi-choice datasets and are thus comparable to each other.

C.2 More variants of the demonstrations

We explored **demonstrations with a constant label** where all labels in the demonstrations are replaced with a constant text, "answer". Specifically, a prediction is made via $\mathrm{argmax}_{y\in\mathcal{C}} P(y|x_1, \texttt{answer}...x_k, \texttt{answer}, x)$. This can be viewed as another way to remove the impact of the label space while keeping the impact of the distribution of the input text. However, results are consistently worse than the results of demonstrations with random English labels. We think this is because constant labels actually change the format of the demonstrations, since they can be viewed as part of a separator between different demonstration examples.

We also explored **demonstrations with the test input** where all inputs in the demonstrations are replaced with the test input, each paired with a random label. Specifically, a prediction is made via $\mathrm{argmax}_{y\in\mathcal{C}} P(y|x, \tilde{y}_1...x, \tilde{y}_k, x)$, where $\tilde{y}_i$ $(1 \leq i \leq k)$ is randomly sampled at uniform from $\mathcal{C}$. This variant is seemingly a reasonable choice given that it satisfies the condition that the inputs in the demonstrations come from the same distribution as the test input (since they are identical), and using random labels is as good as using gold labels. Nonetheless, we find that this variant is significantly worse than most other methods with demonstrations. We think this is because using the constant input for all demonstration example significantly changes the format of the sequence, since the input can be viewed as part of a separator between different demonstration examples.

Dataset	Type	Example
MRPC	Minimal	sentence 1: Cisco pared spending to compensate for sluggish sales . [SEP] sentence 2: In response to sluggish sales , Cisco pared spending . \n {equivalent|not_equivalent}
	Manual	Cisco pared spending to compensate for sluggish sales . \n The question is: In response to sluggish sales , Cisco pared spending . True or False? \n The answer is:{True|False}
RTE	Minimal	sentence 1: The girl was found in Drummondville. [SEP] sentence 2: Drummondville contains the girl. \n {entailment|not_entailment}
	Manual	The girl was found in Drummondville. \n The question is: Drummondville contains the girl. True or False? \n The answer is:{True|False}
Tweet_eval-hate	Minimal	The Truth about #Immigration \n {hate|non-hate}
	Manual	Tweet: The Truth about #Immigration \n Sentiment: {against|favor}
SICK	Minimal	sentence 1: A man is screaming. [SEP] sentence 2: A man is scared. \n {contradiction|entailment|neutral}
	Manual	A man is screaming. \n The question is: A man is scared. True or False? \n The answer is: {False|True|Not sure}
poem-sentiment	Minimal	willis sneered: \n {negative|no_impact|positive}
	Manual	willis sneered: \n The sentiment is: {negative|no_impact|positive}
OpenbookQA	Minimal	What creates a valley? \n {feet|rock|water|sand}
	Manual	The question is: What creates a valley? \n The answer is: {feet|rock|water|sand}
CommonsenseQA	Minimal	What blocks sunshine? \n {summer|park|desktop|sea|moon}
	Manual	The question is: What blocks sunshine? \n The answer is: {summer|park|desktop|sea|moon}
COPA	Minimal	Effect: I coughed. \n {Cause: I inhaled smoke.|Cause: I lowered my voice.}
	Manual	I coughed because {I inhaled smoke.|I lowered my voice.}
ARC	Minimal	Which biome has the most vegetation? \n {desert|forest|grassland|tundra}
	Manual	The question is: Which biome has the most vegetation? \n The answer is: {desert|forest|grassland|tundra}

Table 3: A list of minimal templates taken from Ye et al. (2021); Min et al. (2021b) and manual templates taken from Holtzman et al. (2021); Zhao et al. (2021). Details provided in Appendix B. See Figure 6 for discussion in empirical results. The input and the label are in the red text and in the blue text, respectively. Note that | is used to separate different options for the labels.

Demos w/ gold labels	(*Format* ✓ *Input distribution* ✓ *Label space* ✓ *Input-label mapping* ✓) Circulation revenue has increased by 5% in Finland and 4% in Sweden in 2008. \n positive Panostaja did not disclose the purchase price. \n neutral
Demos w/ random labels	(*Format* ✓ *Input distribution* ✓ *Label space* ✓ *Input-label mapping* ✗) Circulation revenue has increased by 5% in Finland and 4% in Sweden in 2008. \n neutral Panostaja did not disclose the purchase price. \n negative
OOD Demos w/ random labels	(*Format* ✓ *Input distribution* ✗ *Label space* ✓ *Input-label mapping* ✗) Colour-printed lithograph. Very good condition. Image size: 15 x 23 1/2 inches. \n neutral Many accompanying marketing claims of cannabis products are often well-meaning. \n negative
Demos w/ random English words	(*Format* ✓ *Input distribution* ✓ *Label space* ✗ *Input-label mapping* ✗) Circulation revenue has increased by 5% in Finland and 4% in Sweden in 2008. \n unanimity Panostaja did not disclose the purchase price. \n wave
Demos w/o labels	(*Format* ✗ *Input distribution* ✓ *Label space* ✗ *Input-label mapping* ✗) Circulation revenue has increased by 5% in Finland and 4% in Sweden in 2008. Panostaja did not disclose the purchase price.
Demos labels only	(*Format* ✗ *Input distribution* ✗ *Label space* ✓ *Input-label mapping* ✗) positive neutral

Table 4: Example demonstrations when using methods in Section 5. The financial_phrasebank dataset with $\mathcal{C} = \{$"positive", "neutral", "negative"$\}$ is used. Red text indicates the text is sampled from an external corpus; blue text indicates the labels are randomly sampled from the label set; purple text indicates a random English word.