CS 6501 Natural Language Processing

In-context Learning

Yangfeng Ji

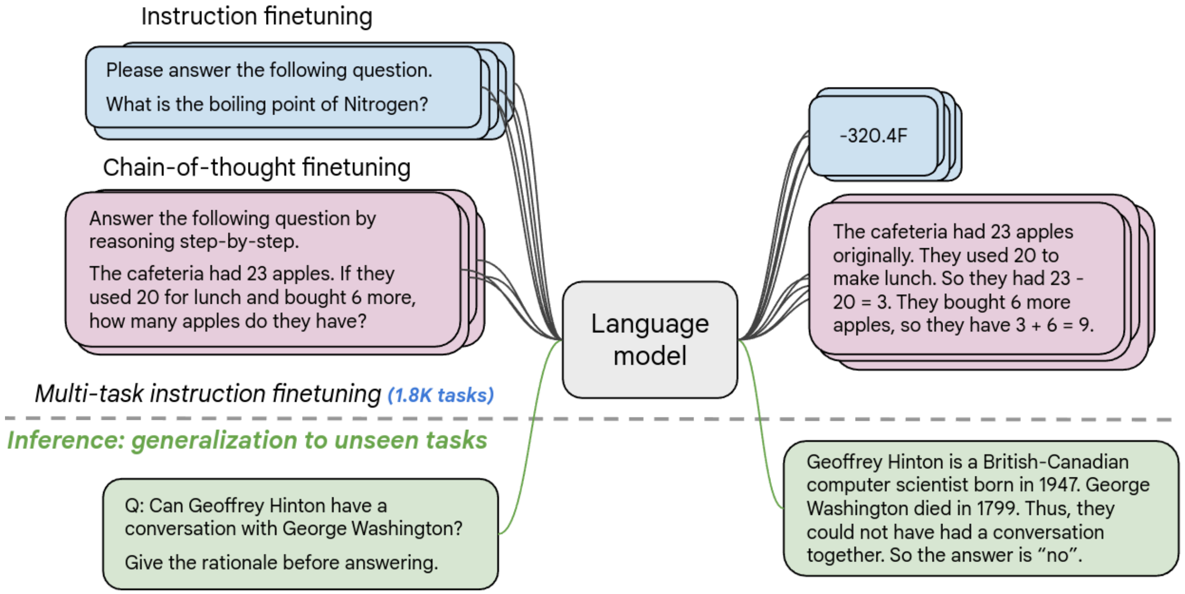
Information and Language Processing Lab
Department of Computer Science
University of Virginia
https://uvanlp.org/

Section I

In-Context Learning

Instruction Tuning

Training language models on a collection of tasks phrased as instructions, which enables models to respond better to instructions and reduces the need for few-shot examples.



Chung et al., 2024

Benchmarking Cross-Task Generalization

Visualization of different instruction-tuning benchmarks

(a) SUP-NATINST (this work)

(b) NATINST

(c) PROMPTSOURCE (T0 subset)

(d) FLAN

(e) INSTRUCTGPT

Wang et al., 2022

Formulation

A simple mathematical formulation of ICL

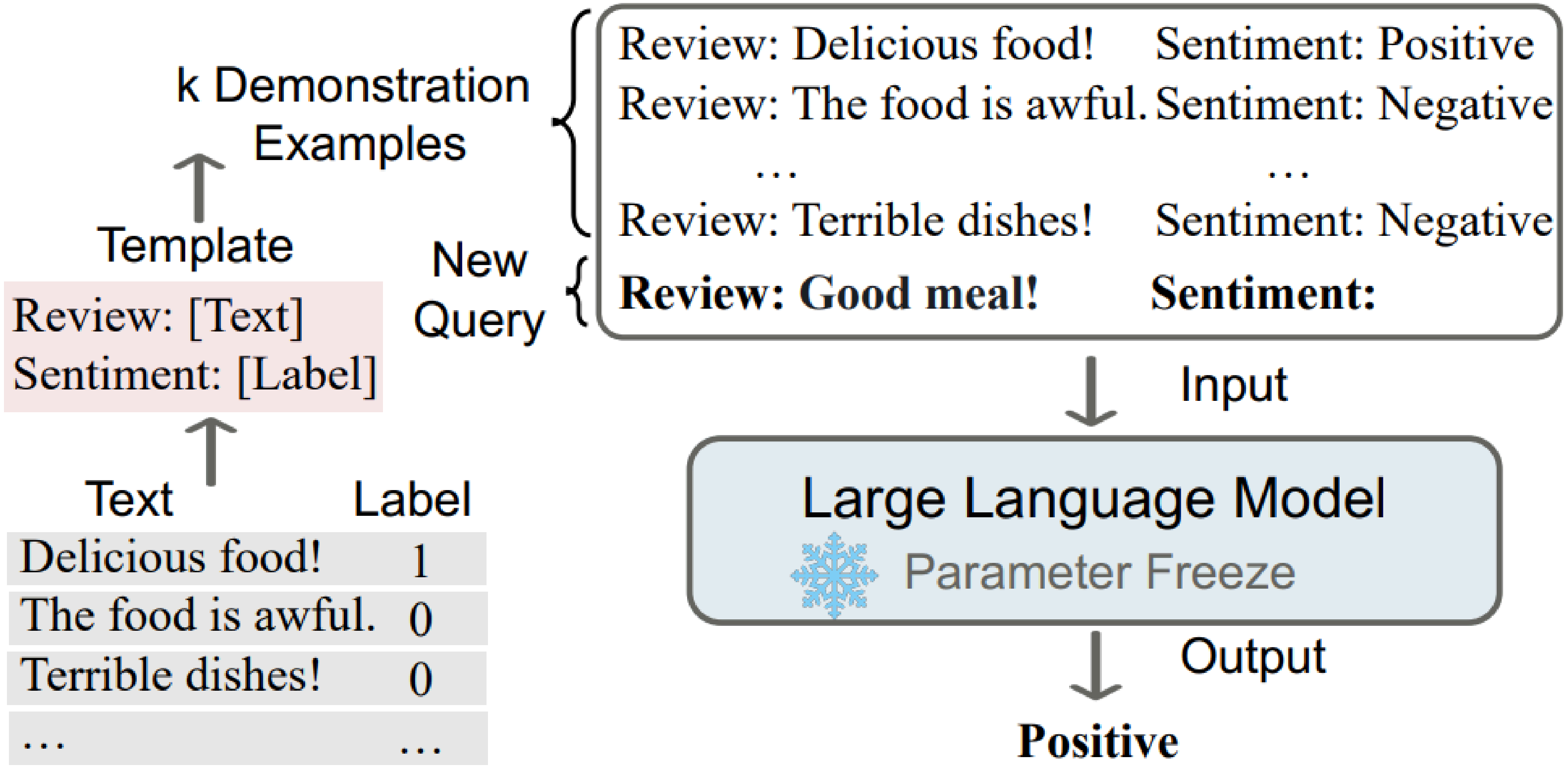
$$P(y_j \mid x) = f_{\mathcal{M}}(y_j, C, x)$$

where

- y_j is the j-th candidate answer, and
- C contains an optional task instruction I and k demonstrations.

Basic Idea

Select a few examples and add them to the prompt as demonstrations



(Dong et al., 2023)

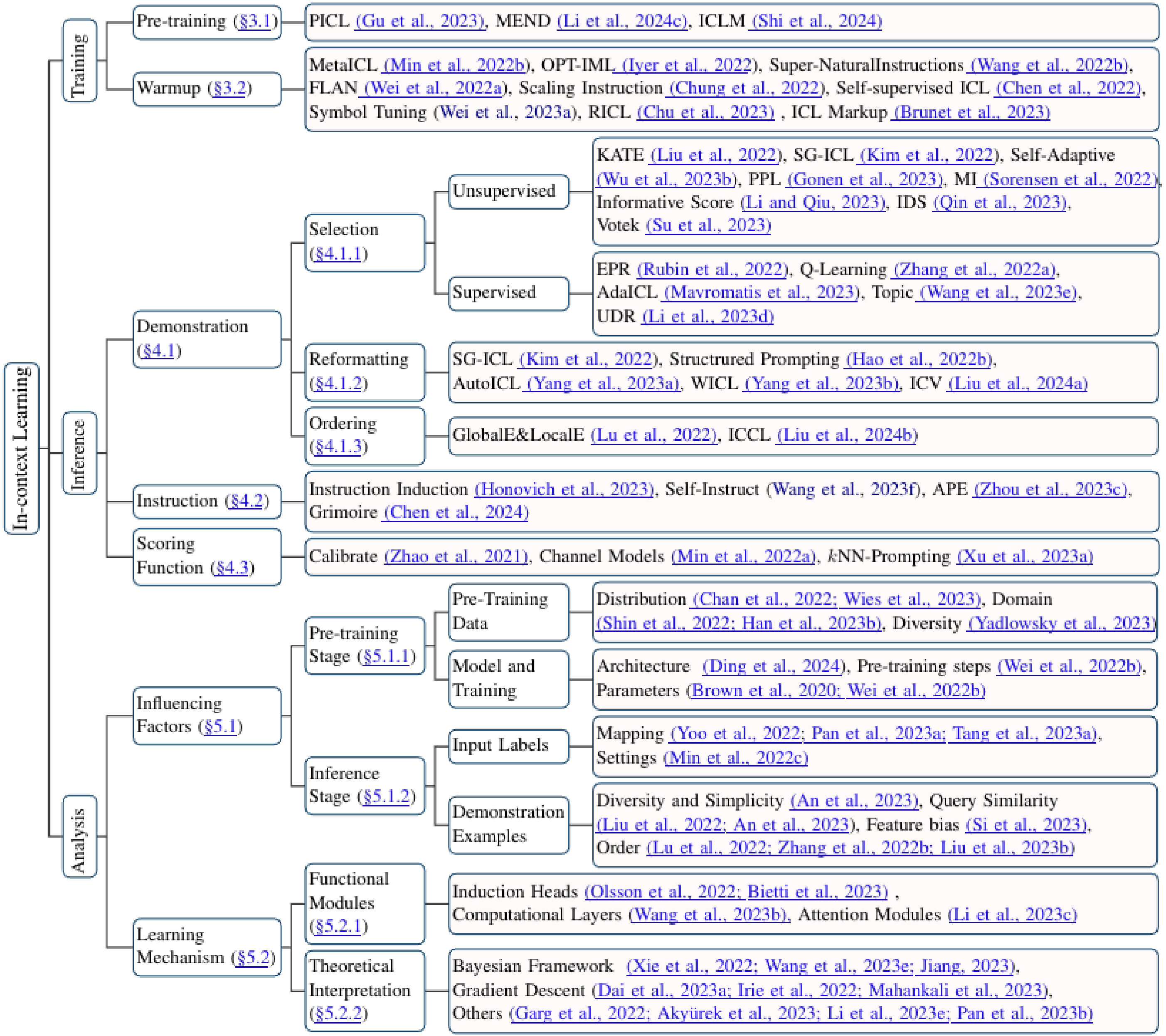
Why In-context Learning

Possible reasons of using in-context learning

- Training-free
- An interpretable way to communicate with users
- Transductive inference vs. inductive inference

Taxonomy of ICL

(Dong et al., 2023)



Related Concepts

- ICL vs. Prompt Tuning
  - ICL is a subclass of prompt tuning
- ICL vs. Few-shot Learning
  - ICL performs few-shot fine-tuning
  - Without parameter update

Supervised ICL

ICL can be done by explicitly fine-tuning the model to follow the format

	Meta-training	Inference
Task	C *meta-training* tasks	An unseen *target* task
Data given	Training examples $\mathcal{T}_i = \{(x_j^i, y_j^i)\}_{j=1}^{N_i}, \ \forall i \in [1, C] \quad (N_i \gg k)$	Training examples $(x_1, y_1), \cdots, (x_k, y_k)$, Test input x
Objective	For each iteration, 1. Sample task $i \in [1, C]$ 2. Sample $k+1$ examples from $\mathcal{T}_i$: $(x_1, y_1), \cdots, (x_{k+1}, y_{k+1})$ 3. Maximize $P(y_{k+1} \| x_1, y_1, \cdots, x_k, y_k, x_{k+1})$	$\text{argmax}_{c \in \mathcal{C}} P(c \| x_1, y_1, \cdots, x_k, y_k, x)$

Table 1: Overview of MetaICL (Section 3). MetaICL uses the same in-context learning setup at both meta-training and inference. At meta-training time, $k+1$ examples for a task is sampled, where the last example acts as the test example and the rest k examples act as the training examples. Inference is the same as typical in-context learning where k labeled examples are used to make a prediction for a test input.

Section II

Demonstration Construction

Select Similar Examples

E.g., select the k-nearest neighbors in the latent space

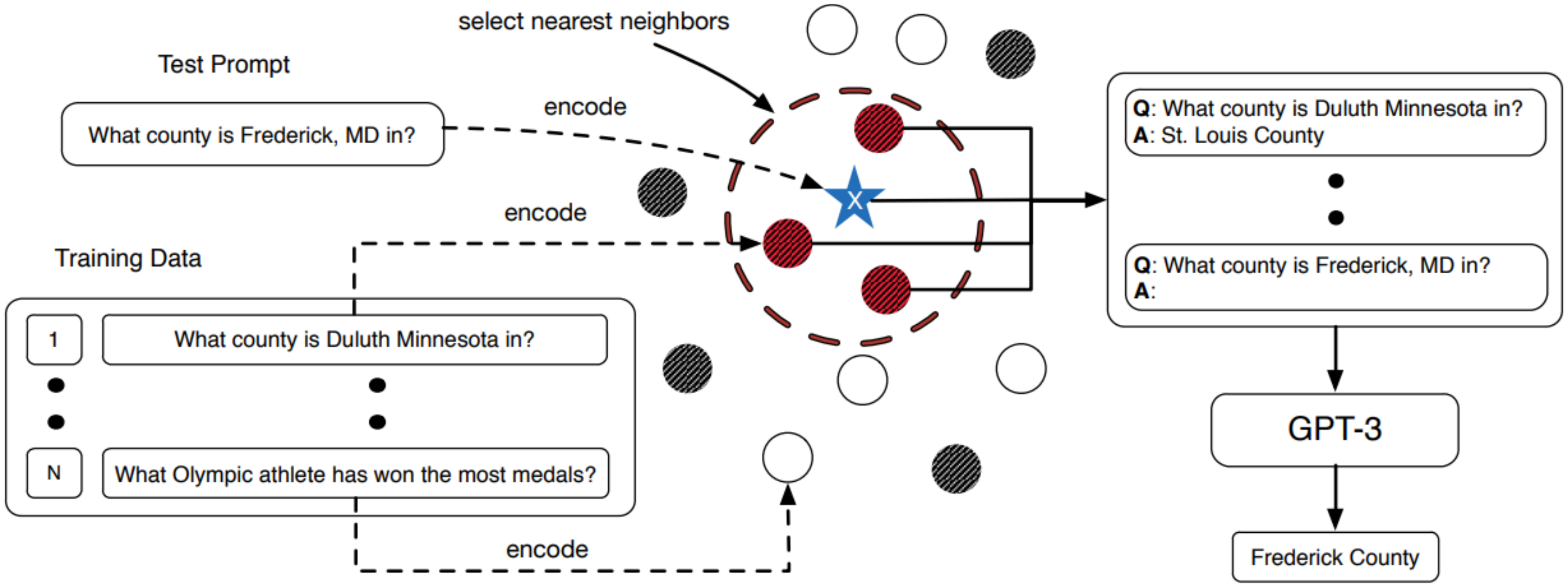
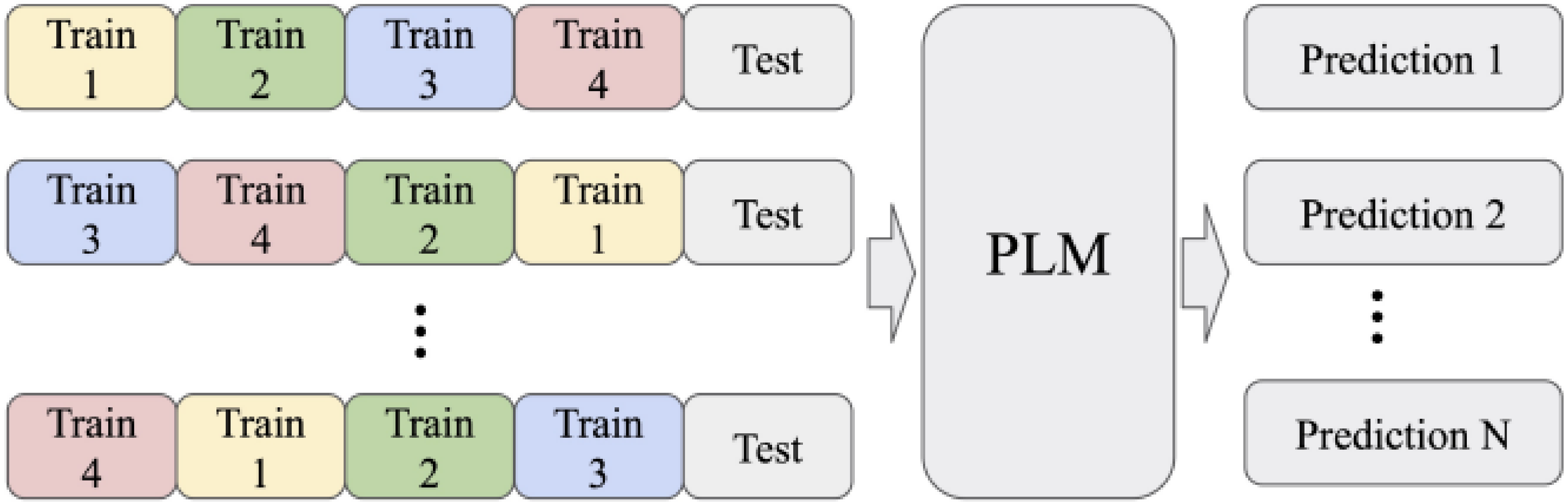


Figure 1: In-context example selection for GPT-3. White dots: unused training samples; grey dots: randomly sampled training samples; red dots: training samples selected by the k-nearest neighbors algorithm in the embedding space of a sentence encoder.

(Liu et al., 2022)

Example Order

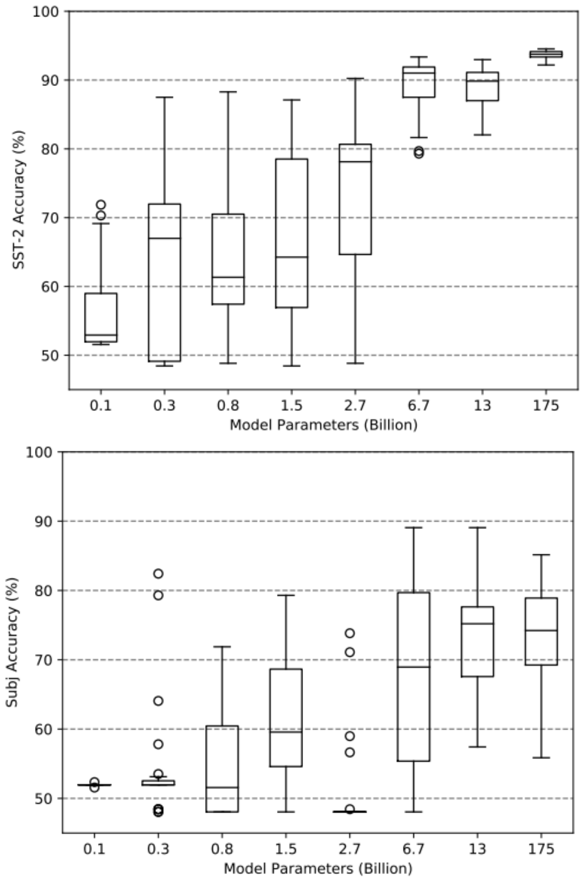
Permute the in-context examples



(Lu et al., 2022)

Example Order: Performance Difference

The impact depends on the tasks and the model sizes



Section III

ICL Explanation

Rethinking the Role of Demonstrations

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

Link

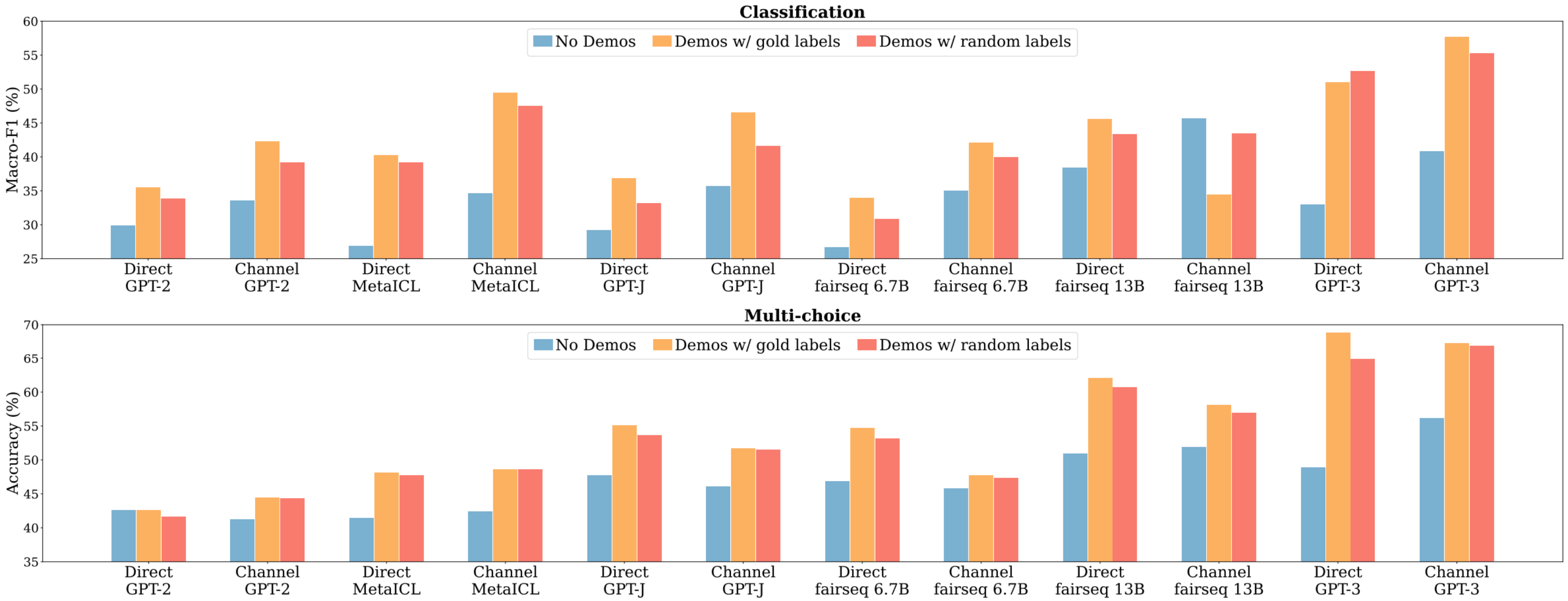
Experiment Setup

- 12 models

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‡	✗	✗

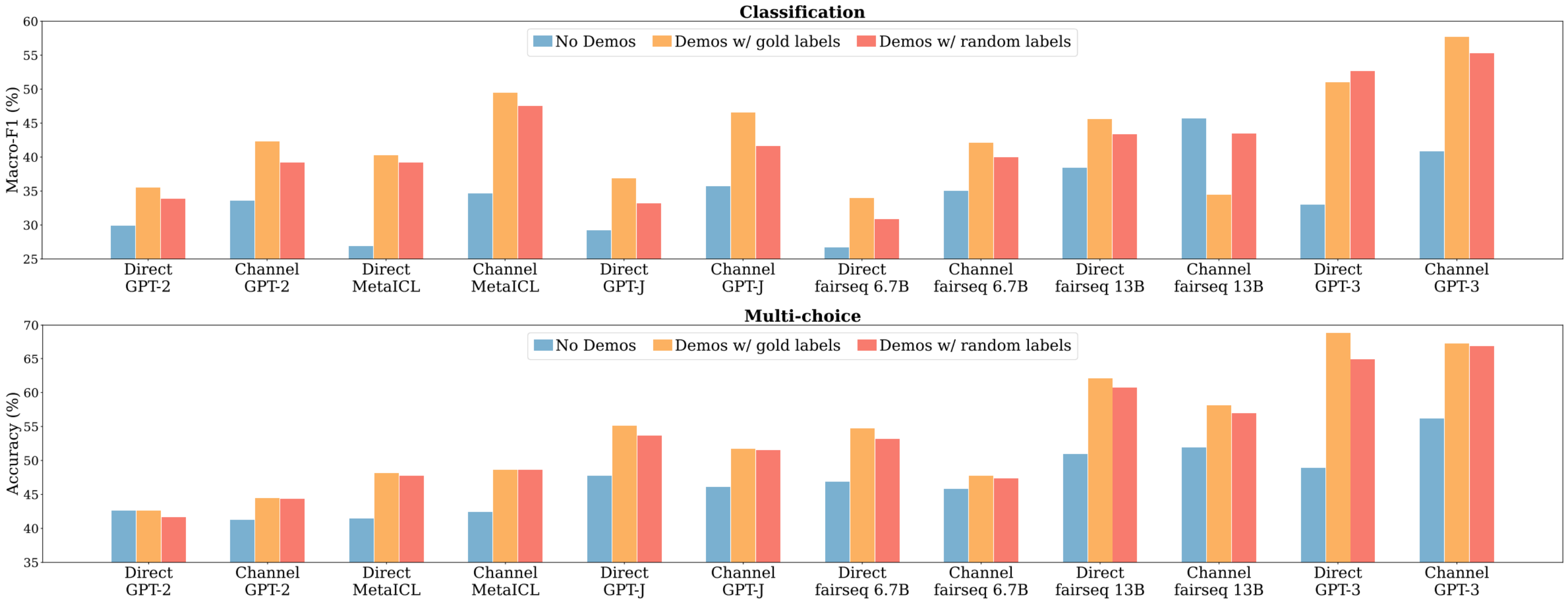
- 26 datasets
- $k = 16$ examples as demonstrations

Ground Truth Matters Little



- Left: no demonstration
- Middle: demonstrations with ground-truth labels
- Right: demonstrations with random labels

What Impacts ICL?



Models learn something from ICL, but *"it is not directly from the pairings in the demonstrations"*.

What about MetaICL?

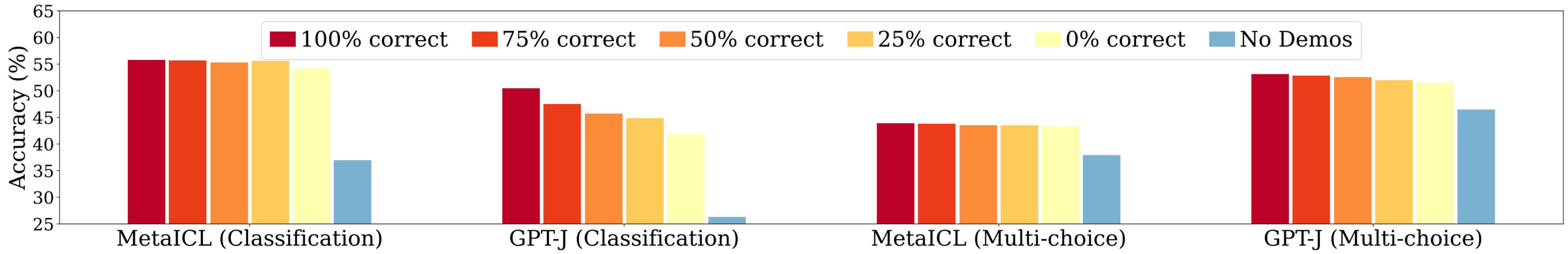
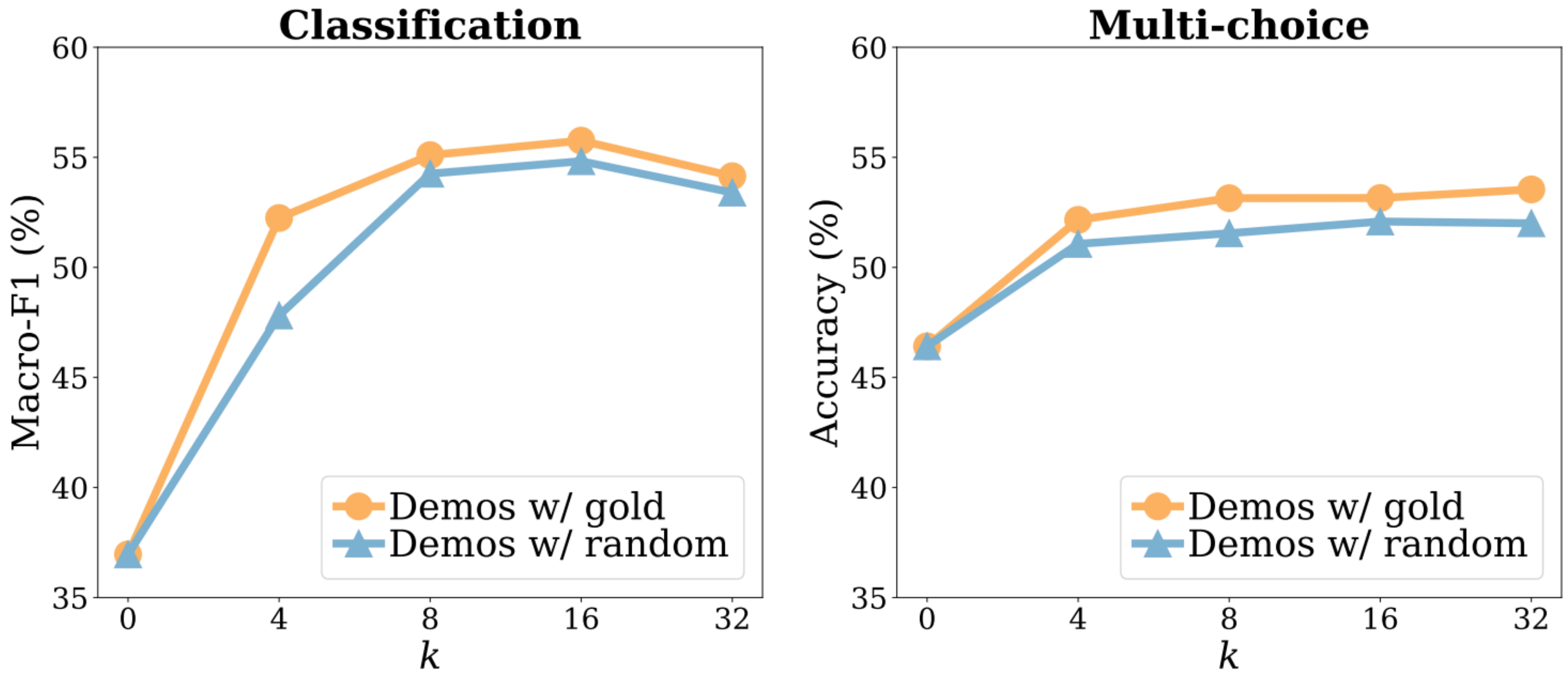


Figure 4: Results with varying number of correct labels in the demonstrations. Channel and Direct used for classification and multi-choice, respectively. Performance with no demonstrations (blue) is reported as a reference.

MetaICL with an explicit ICL training objective actually encourages the model to **ignore** the input-label mapping.

What about with Different k's?

The pattern is the same with different numbers of demonstrations (k)

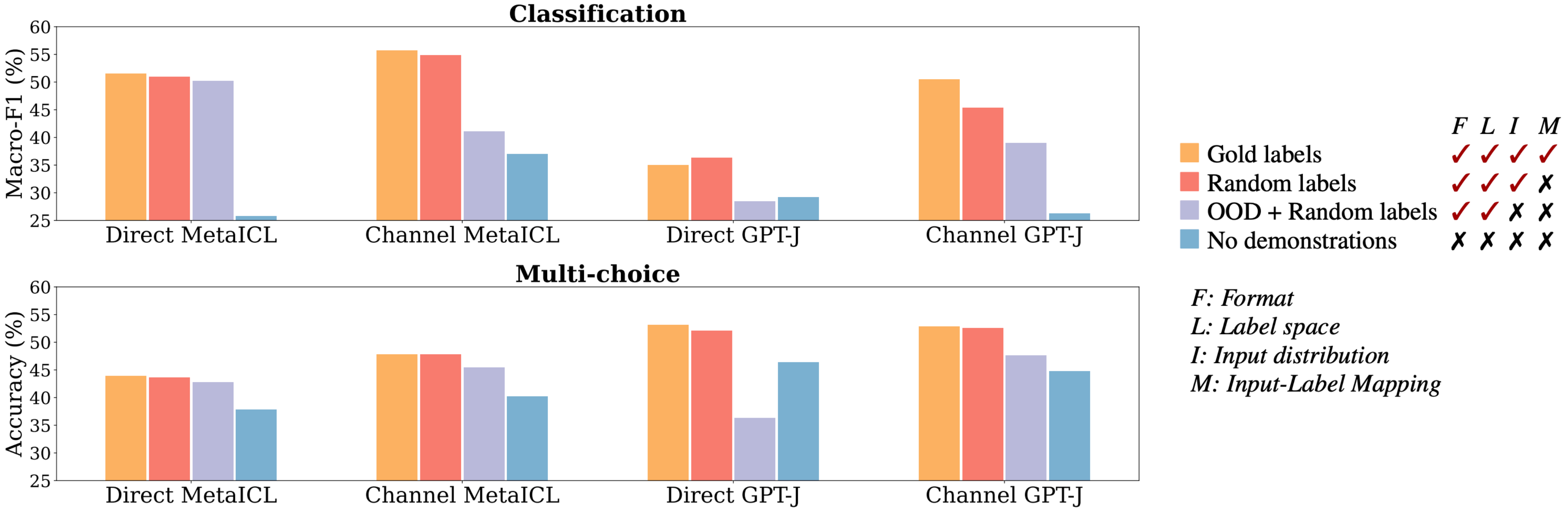


Potential Reasons for ICL

- The distribution of input text
  - About input
- The label space
  - About label
- The input-label mapping
  - This is how supervised learning works
- The format

About Input Distribution

Using out-of-distribution examples as a comparison



Label Space

For comparison purpose, replace labels with random English words

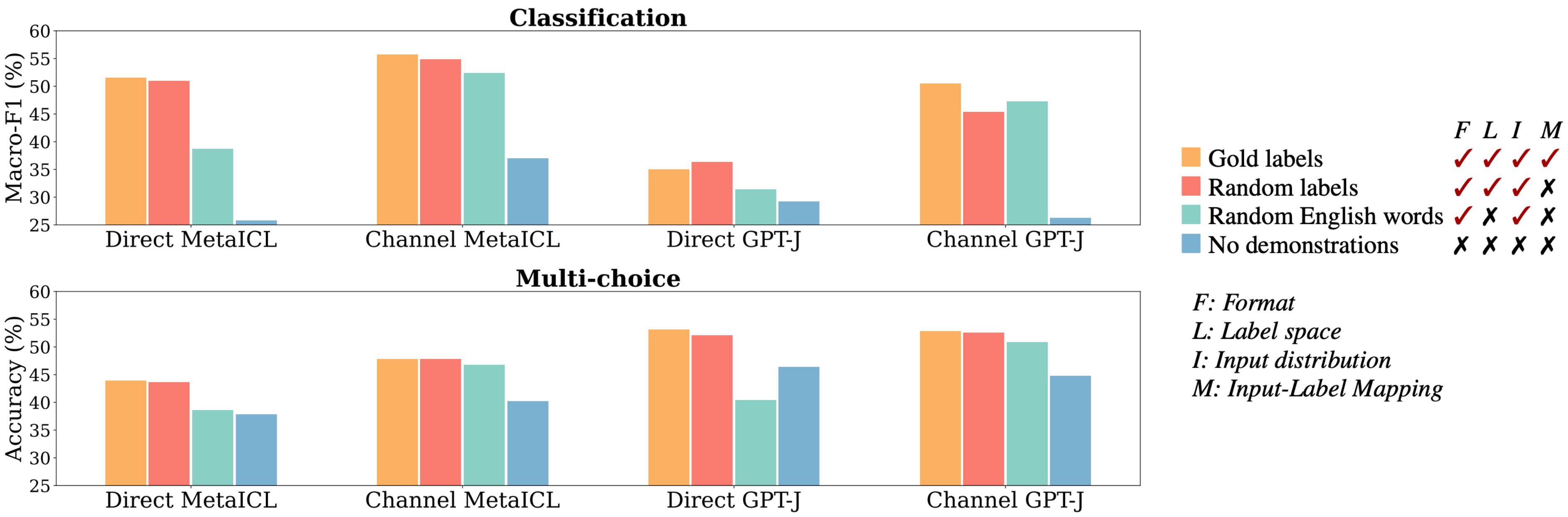


Figure 9: Impact of the label space. Evaluated in classification (top) and multi-choice (bottom). The impact of the label space can be measured by comparing ■ and ■. The gap is significant in the direct models but not in the channel models (discussion in Section 5.2).

Input-label Pairing Format

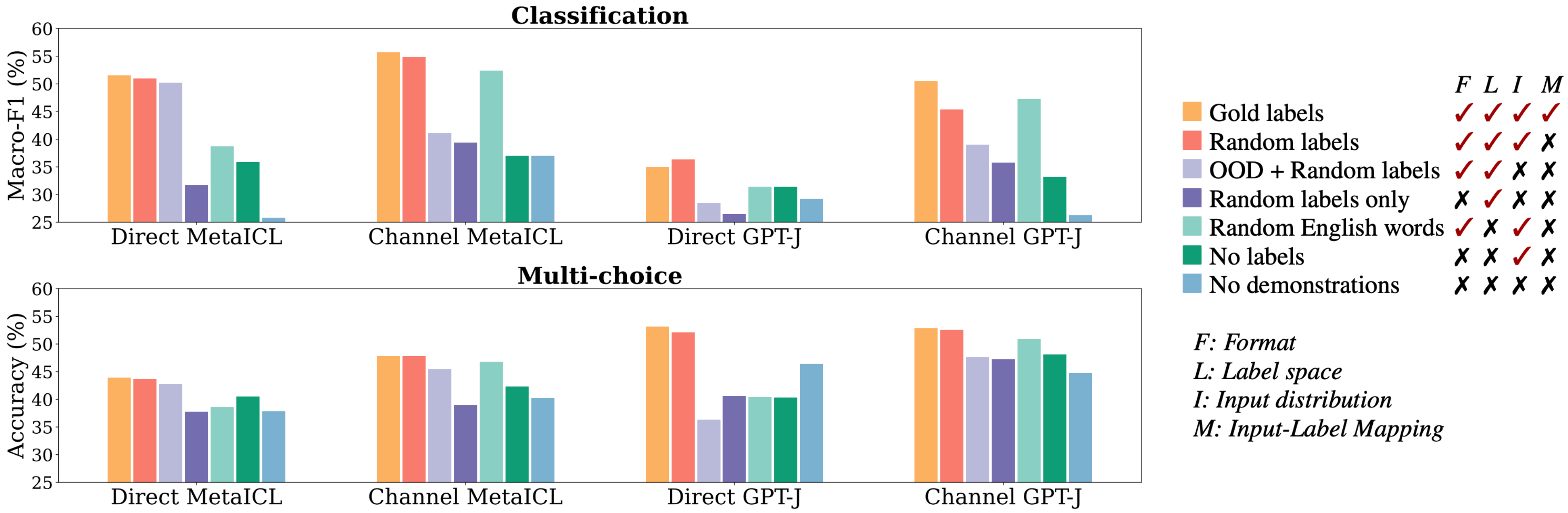


Figure 10: Impact of the format, i.e., the use of the input-label pairs. Evaluated in classification (top) and multi-choice (bottom). Variants of demonstrations without keeping the format (■ and ■) are overall not better than no demonstrations (■). Keeping the format is especially significant when it 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.

Takeaway

- It works
- We have some ideas about *how* to make it work
- We have little ideas about *why* it works

Thank You!