Exploring Distributional Shifts in Large Language Models for Code Analysis

Shushan Arakelyan[1], Rocktim Jyoti Das[2], Yi Mao[3], Xiang Ren[1]

[1]University of Southern California, [2]IIT Delhi, [3]Microsoft Azure AI

Motivation

20 years ago…

"What characteristics differ between projects used for building predictors?"

Open source	Yes/No
Global development	Yes/No
Code reviews	Yes/No
Static checkers	Yes/No
…	

Today

More data ✔

Larger models ✔

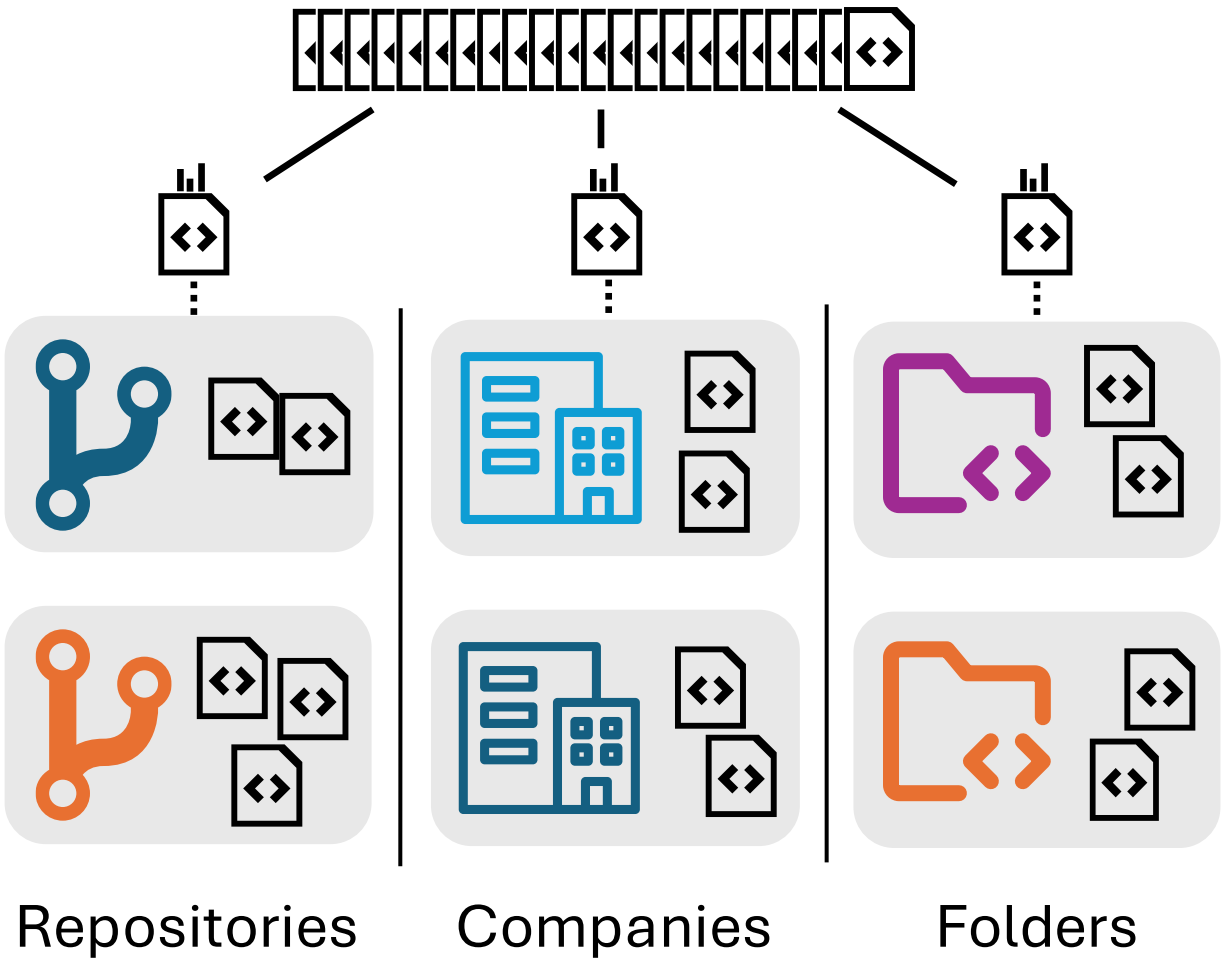
More deployment ✔

Same challenges?

[1] Turhan, B. "On the dataset shift problem in software engineering prediction models"

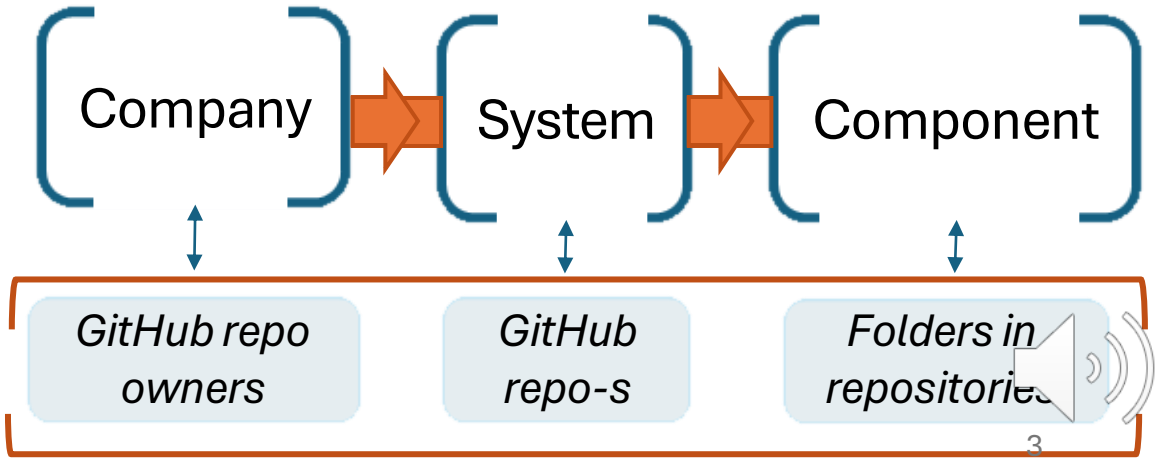
[2] Zimmerman, T., et al "Cross-project Defect Prediction: A Large Scale Experiment on Data vs. Domain vs. Process"

Data preparation



CODE SEARCH	TRAINING	DEV	TESTING
GO	635,635	28,483	14,291
JAVA	908,886	30,655	26,909
JAVASCRIPT	247,773	16,505	6,483
PHP	1,047,406	52,029	28,391
PYTHON	824,342	46,213	22,176
RUBY	97,580	4,417	2,279

	Train
org.	9737
repos.	15858
fold.	25268

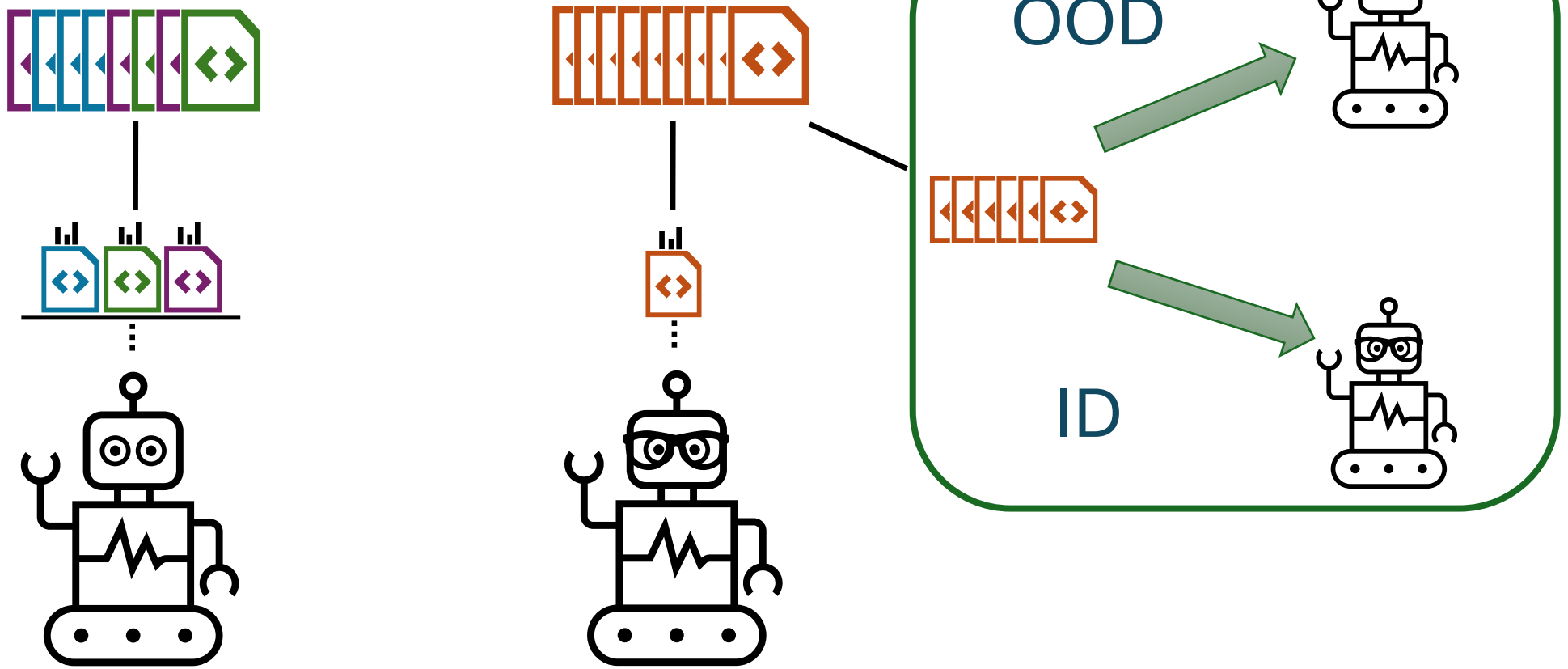


Experimental setup

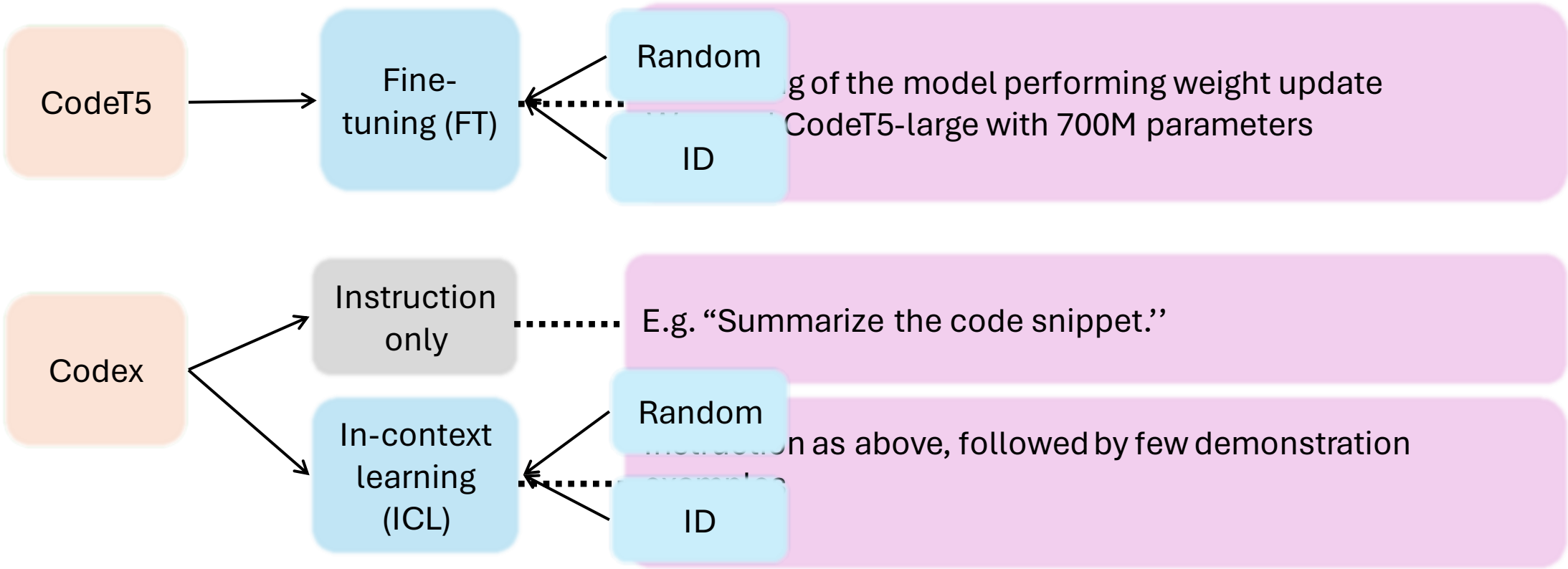
Train data

Target domain

Test both models on unseen samples from the target domain



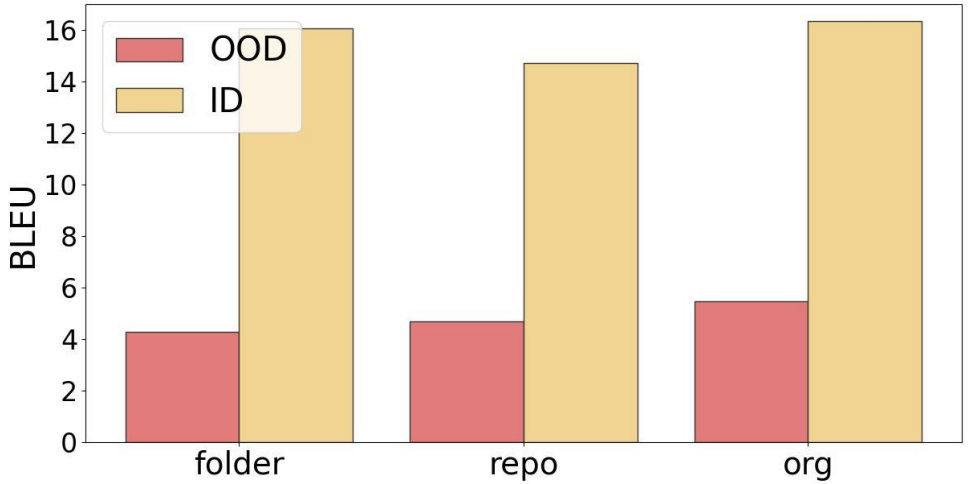
Models and methods



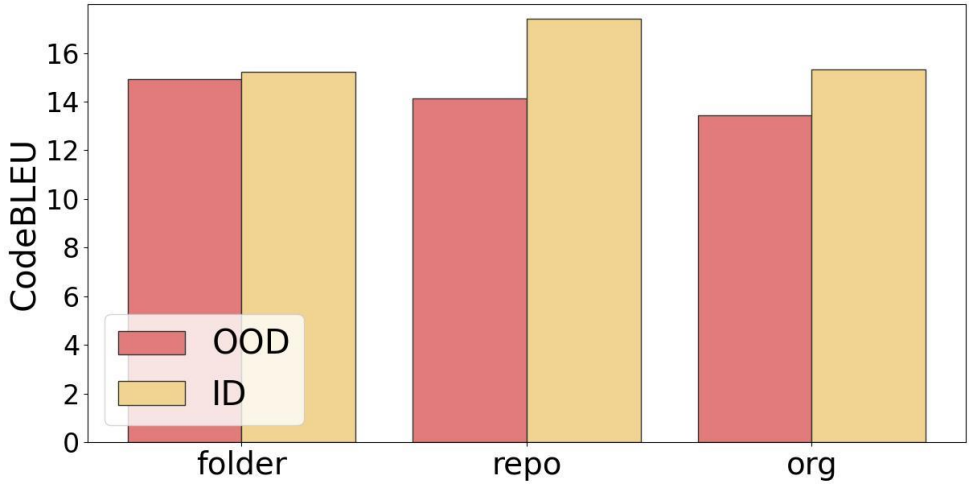
[1] Hu et al., “Lora: Low-rank adaptation of large language models”

Results: Performance ID vs OOD

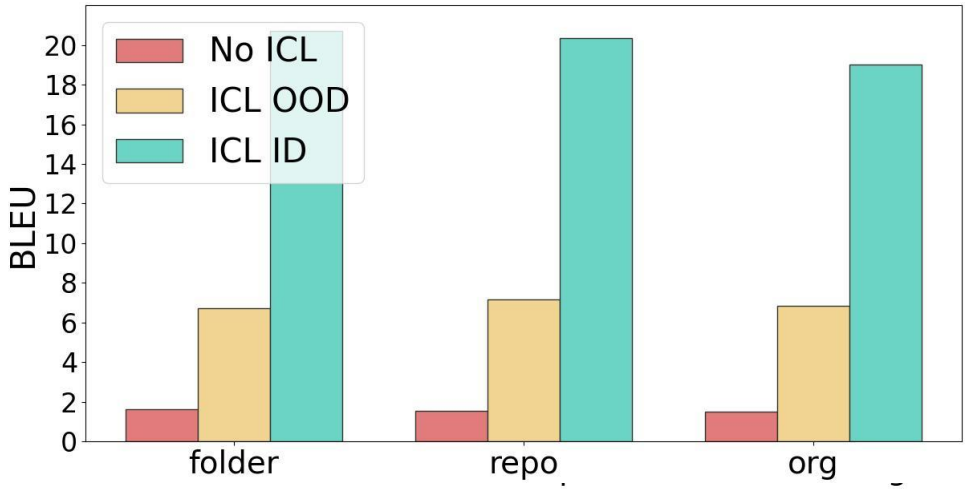
Code summarization

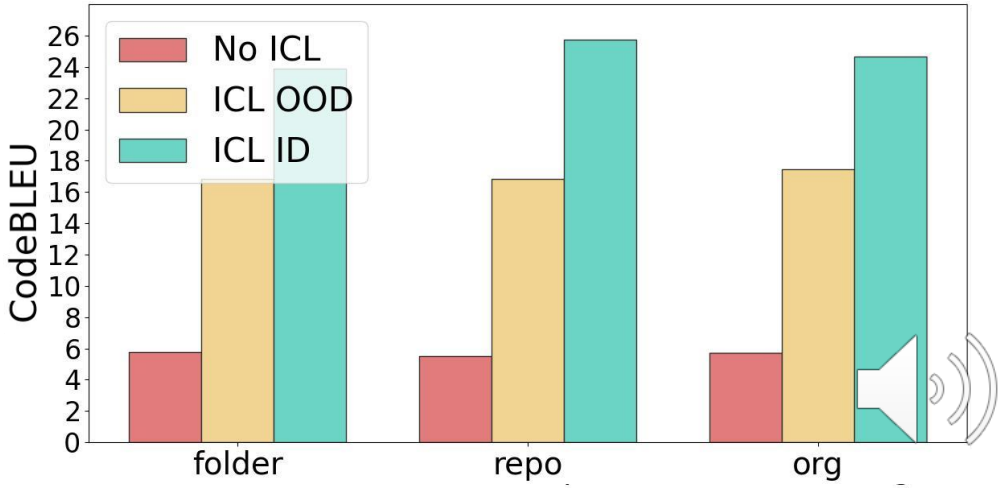


Code generation



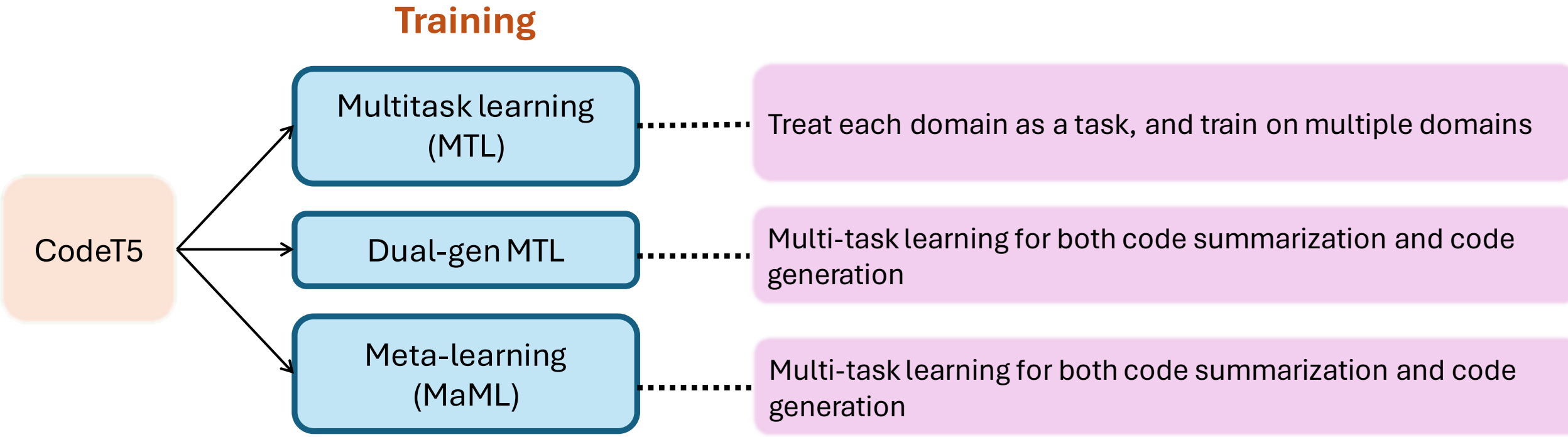
CodeT5



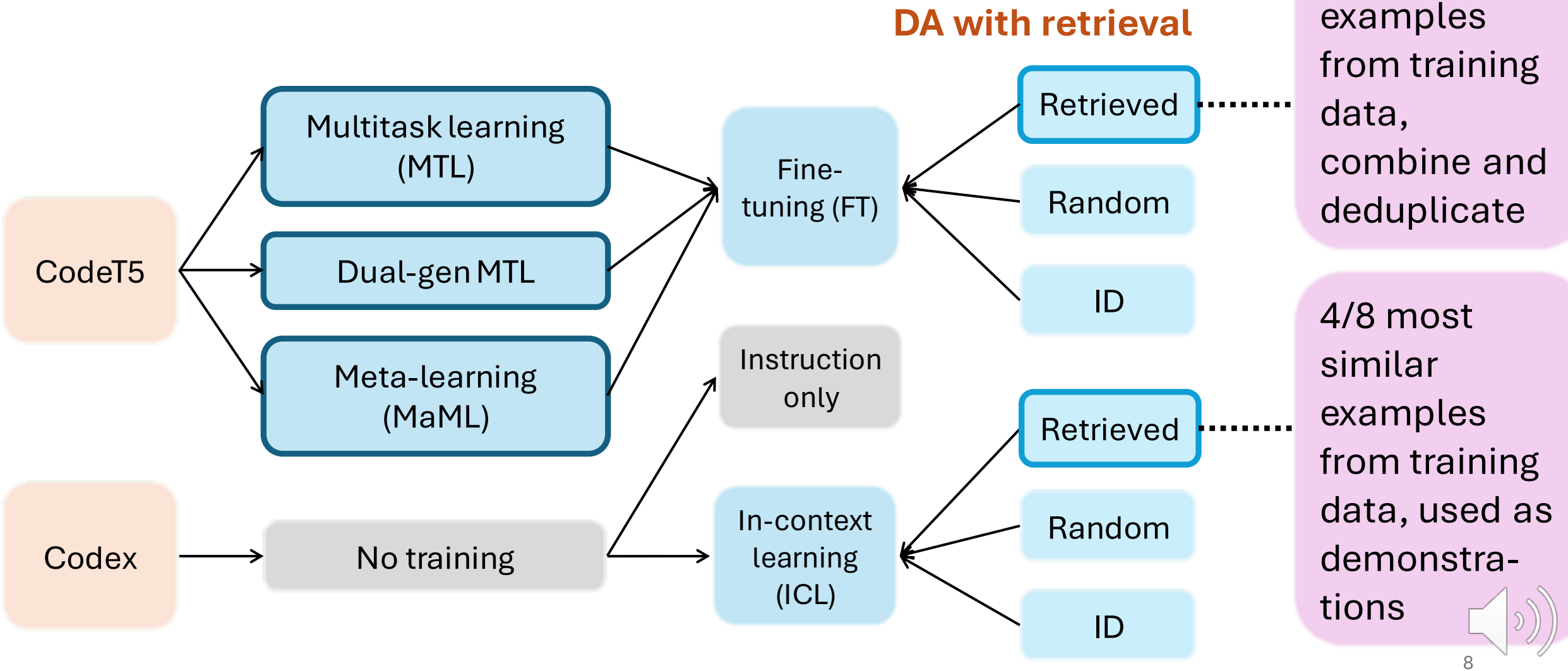


Codex

How to improve OOD performance?



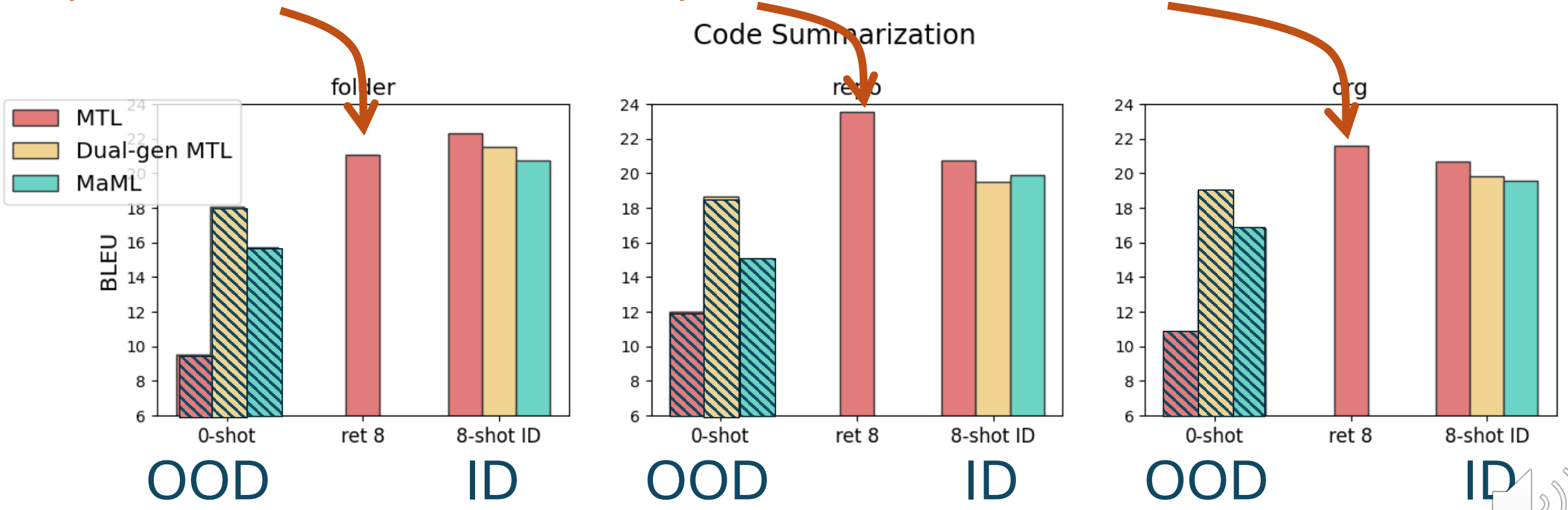
How to improve OOD performance?



Results [CodeT5]

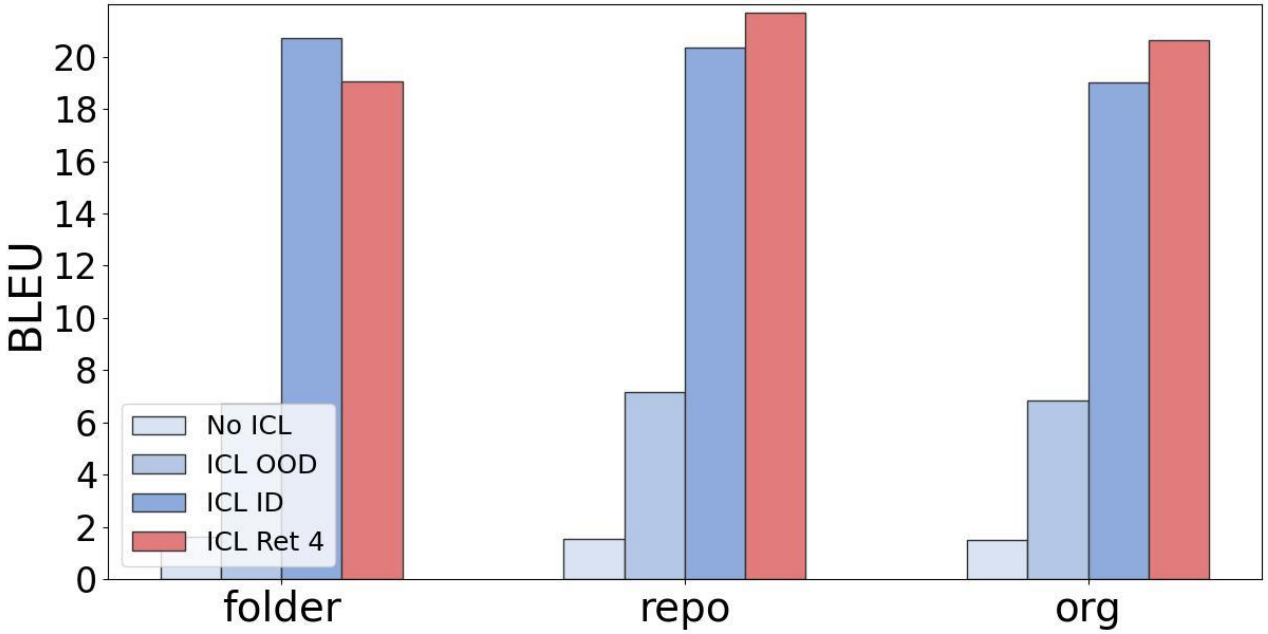
Training does not get rid of ID vs OOD performance discrepance

Supervision with retrieved examples is more effective!

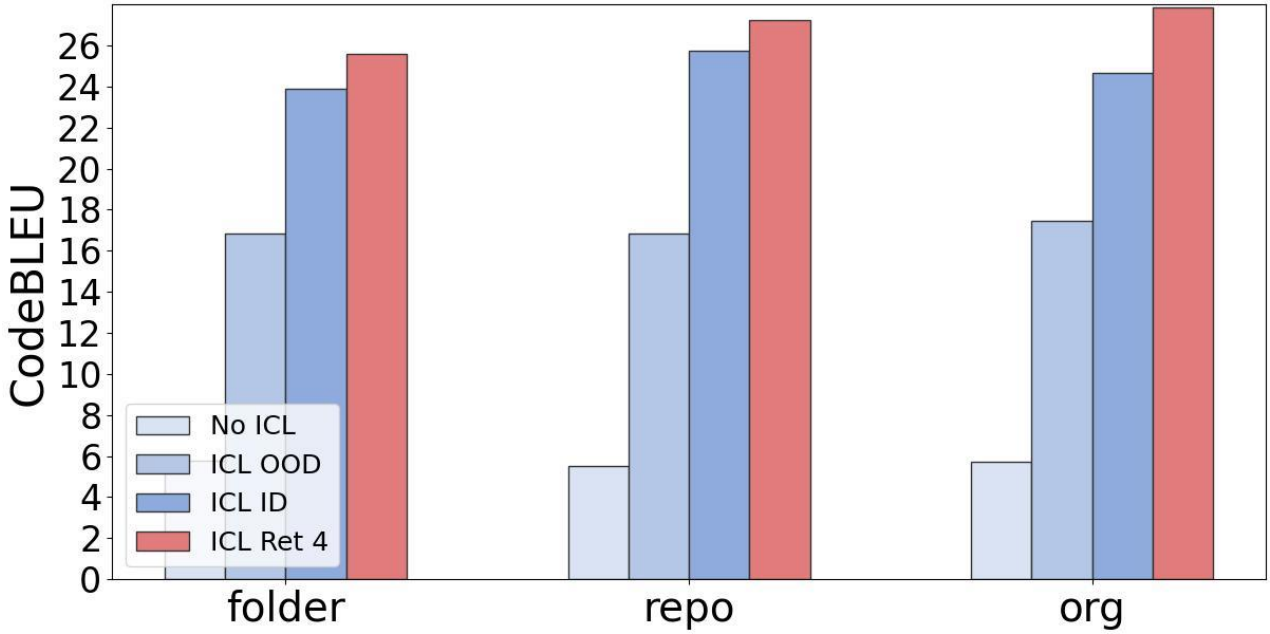


Results [Codex]

Code summarization



Code generation



Supervision with retrieved examples is effective with ICL

Findings

- Splits naturally occurring in software present distributional shift challenge
- Domain adaptation can be effective with a very small amount of data
- Retrieving examples for supervision is effective in combating distribution shift