Data Distributional Properties Drive Emergent In-Context Learning in Transformers

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, Jay McClelland, Felix Hill
Two distinct modes of learning in neural networks

In-weights learning
- gradient-based
- slow: needs many examples
- standard supervised learning

In-context learning
- no gradient updates
- rapid: from a few examples
- few-shot / one-shot learning

Normally, only happens if we explicitly meta-train for it:

E.g.: Santoro et al, 2016; Vinyals et al, 2016; Wang et al, 2016
But in-context (few-shot) learning can also emerge ...when trained on a very different objective!

Language Models are Few-Shot Learners

```
GPT-3

sea otter => loutre de mer
peppermint => menthe poivrée
plush giraffe => girafe peluche
cheese => ........................................
```

Flamingo

"I have a couple of other projects I'm excited about."
How do large transformer models achieve *emergent* in-context learning?

Hypothesis: Maybe it’s because the **distributions of naturalistic data** have special properties.

**Natural data is long-tailed**

![Long-tailed distribution graph](image1)

**Natural data is bursty**

![Bursty distribution graph](image2)

Maybe training on naturalistic data is like an interpolation between supervised and few-shot meta-training...
few-shot learning emerges, even without explicit training

**Standard Supervised**
- items recur and are uniform
- label mappings are fixed

![Images of fruits and arrows indicating "apple" and "banana" with corresponding labels]

**Naturalistic data (e.g. language)**
- words do recur
- word meanings are somewhat fixed

but also:
- rare words do not recur often
- some rare words are bursty
- many-to-many relationships

**Few-shot meta-training**
- explicitly train for few-shot learning
- items differ on every episode
- label mappings are only fixed within episodes

![Images indicating "dax" and "bax" with additional symbols, and a question mark indicating uncertainty]

"few-shot learning emerges, even without explicit training"
Our Project

**Hypothesis:** Certain non-uniformities in data distributions can lead to emergent few-shot (in-context) learning, and this is a general phenomenon.

**Experiments:** Modify a standard few-shot learning image dataset (Omniglot), to control these distributional properties and measure their effects on few-shot learning.

**Implications:**
- understanding how we might design or collect datasets to achieve in-context learning in domains outside of language
General structure of the experiments

transformer (causal)

resnet embed

image label

context

query

?
Training data

- labels are fixed across all of training

Example "bursty" sequence

Two ways to solve:
1. In-weights memorization
2. In-context learning
Evaluation data

Example evaluation sequence for in-context learning
- Two holdout classes, randomly assigned to labels [0, 1]

Example evaluation sequence for in-weights memorization
- The query class was seen in training, and does not appear in the context.
What kinds of training data promote in-context learning?
Importance of **burstiness** in the data

**Train**

- **bursty**
  - b a a b h a g b a
- **non-bursty**
  - f h d b a e g c a

- vary the proportions in the training data

**Eval**

- More burstiness leads to better in-context learning
- In-context learning trades off against weights-based learning
Transformers succeed at the Omniglot challenge: Importance of **number of classes** in the data

More training classes leads to better in-context learning

Again, in-context learning trades off against weights-based learning
Other natural data-inspired distributional properties

Dynamic meaning:

Multiplicity of item-label mappings

![Graph showing the effect of label multiplicity on accuracy.](image)

Within-class variation

![Graphs showing the effect of pixel noise on accuracy.](image)

Figure 4: Dynamic meanings improve in-context learning. Increasing the number of labels per class ('label multiplicity') increases in-context learning.
Can in-weights memorization and in-context learning co-exist in the same model?
We can achieve both kinds of learning when we train on \textit{skewed} distributions.

\begin{align*}
p(X = x) \propto \frac{1}{x^\alpha}
\end{align*}

Intriguingly, Zipf exponent 1 corresponds approximately to the skew in natural languages.

There is a sweet spot at Zipf exponent = 1, where we attain both few-shot learning and in-weights memorization.

Intriguingly, Zipf exponent 1 corresponds approximately to the skew in natural languages.
But architecture does matter too...
Transformers vs RNNs

In-context learning on holdout classes.

(a) Transformer. (b) Vanilla RNN. (c) LSTM.

- each line is a hyperparameter setting
- matched on: # params, # layers, hidden size, training data

➔ Recurrent models never achieve few-shot learning, with the same training data
➔ But even though architecture matters, it’s not enough – we need the right data, too
Implications for compositionality

We study in-context few-shot learning, which can be construed as a narrow instantiation of compositionality.

Our findings on the drivers of this emergent behavior:

- Large-scale data and models are not necessary
- Certain distributions of training data promote it
  - these distributional features are present in natural data like language
- Architecture matters too
  - Transformers > RNNs

- Andreas 2020
- Akyürek & Andreas, 2022
Implications for compositionality

We study in-context few-shot learning, which can be construed as a narrow instantiation of compositionality.

Our findings on the drivers of this emergent behavior:

- Large-scale data and models are not necessary
- Certain distributions of training data promote it
  - these distributional features are present in natural data like language
- Architecture matters too
  - Transformers > RNNs

RNNs with memory augmentation *could* perform well on SCAN (Lake et al 2019)
  - memory-augmented NNs help meta-learning because they have both long-term and short-term storage (that is stable + element-wise addressable) (Santoro et al 2013)

Transformers can perform well on certain kinds of composition

- transformers have the desired properties as well
- transformers may perform compositional operations (Elhage et al 2021; Olsson et al 2022)
Implications for compositionality

We study in-context few-shot learning, which can be construed as a narrow instantiation of compositionality.

Our findings on the drivers of this emergent behavior:

- Large-scale data and models are not necessary
- Certain **distributions of training data** promote it
  - these distributional features are present in natural data like language
- **Architecture** matters too
  - Transformers > RNNs

We need both!

Non-uniformity is important