Strategy Coopetition explains the emergence and transience of in-context learning

Aaditya K Singh, Ted Moskovitz, Sara Dragutinović, Felix Hill, Stephanie CY Chan*, Andrew M Saxe*

In-context learning (ICL)

The ability of a model to learn from inputs at test time to adapt its behavior

- Often contrasted with "in-weights learning" (IWL)

Found to emerge in transformers trained on internet-scale corpora (GPT3)

Chan et al. (2022) found that *data properties* may drive ICL emergence…

- Example: burstiness – a rare word is more likely to occur many times in one document, as opposed to uniformly across documents

… but Singh et al. (2023) (and many others) found *emergent ICL may be transient!*

→ **Why does ICL emerge, if only to fade away later in training?**

We found some surprising things

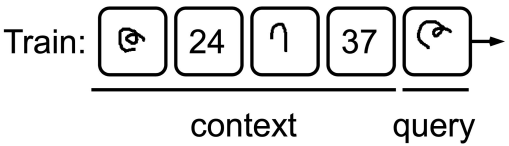
1. The asymptotic strategy (after ICL disappears) is *not* pure IWL, but rather a **mix between ICL + IWL**, which we call "context-constrained IWL" (CIWL)

2. ICL and CIWL **share subcircuits**

3. ICL and CIWL both compete (leading to transience) and cooperate (enabling ICL emergence) → **strategy coopetition**

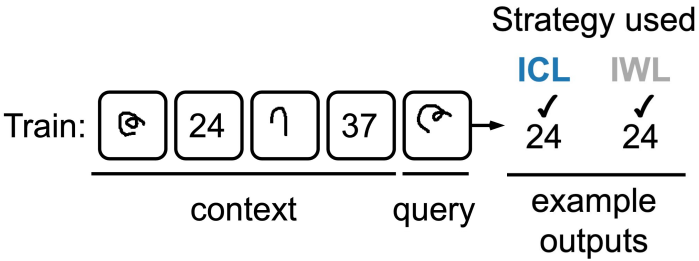
Reproducing the transience effect, at small scale

Train 2L attention-only transformers on sequences of (omniglot image, label) pairs



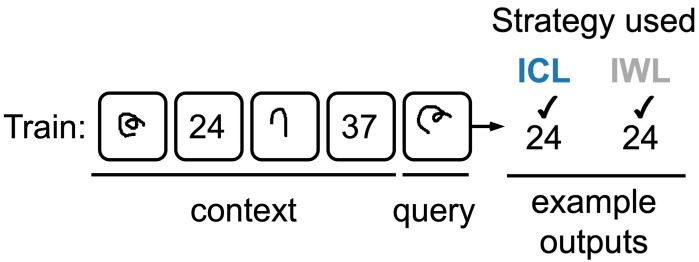
Reproducing the transience effect, at small scale

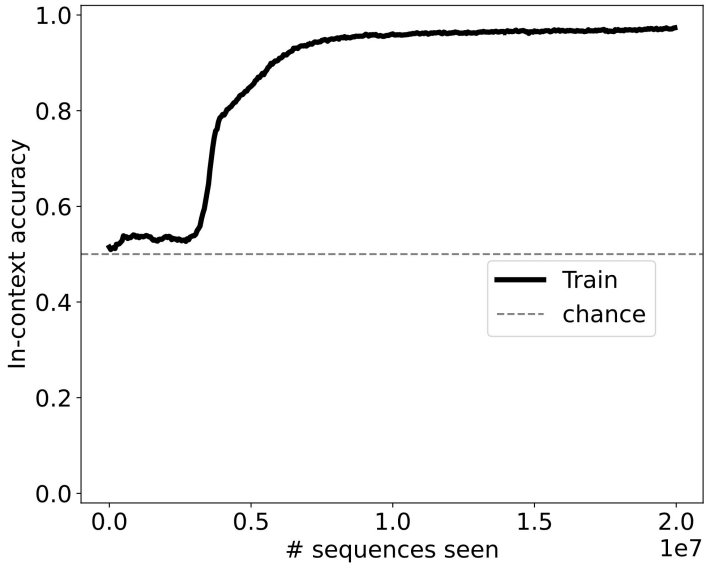
Train 2L attention-only transformers on sequences of (omniglot image, label) pairs



Reproducing the transience effect, at small scale

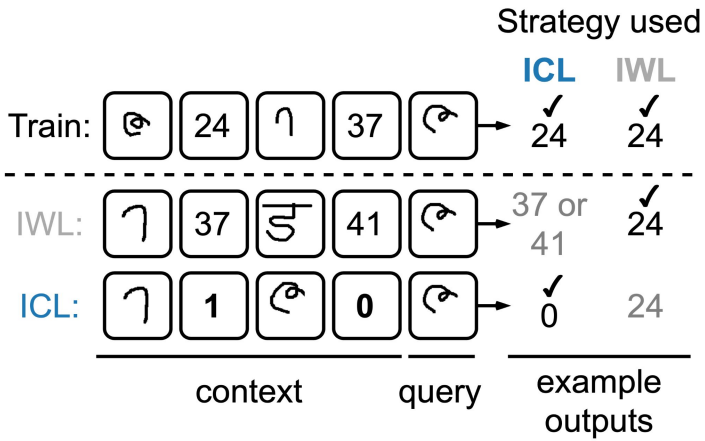
Train 2L attention-only transformers on sequences of (omniglot image, label) pairs

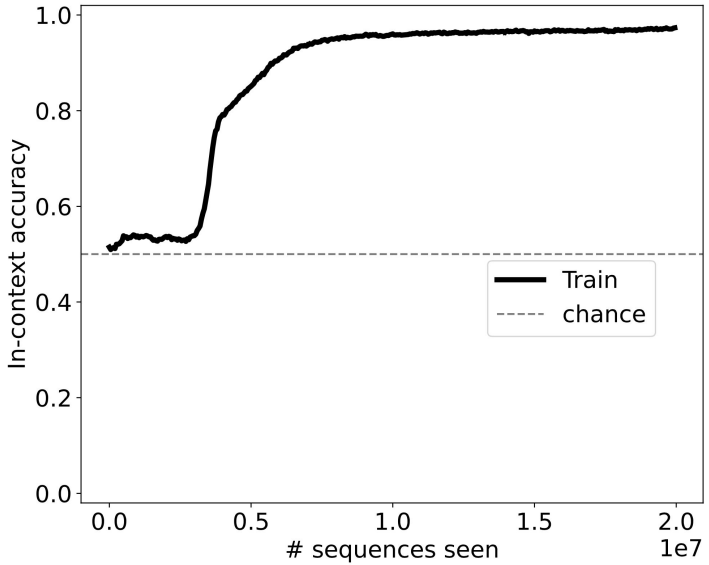




Reproducing the transience effect, at small scale

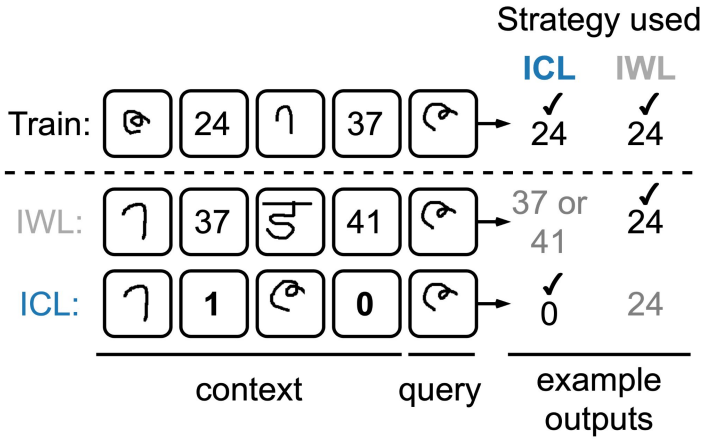
Train 2L attention-only transformers on sequences of (omniglot image, label) pairs

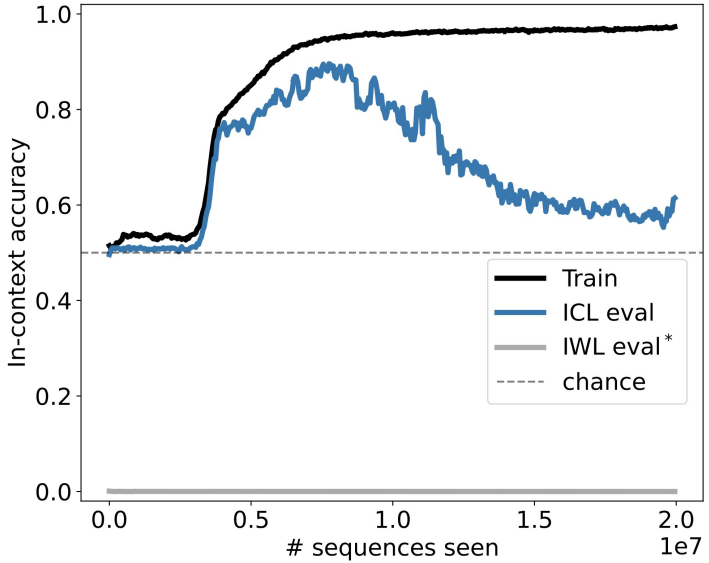




Reproducing the transience effect, at small scale

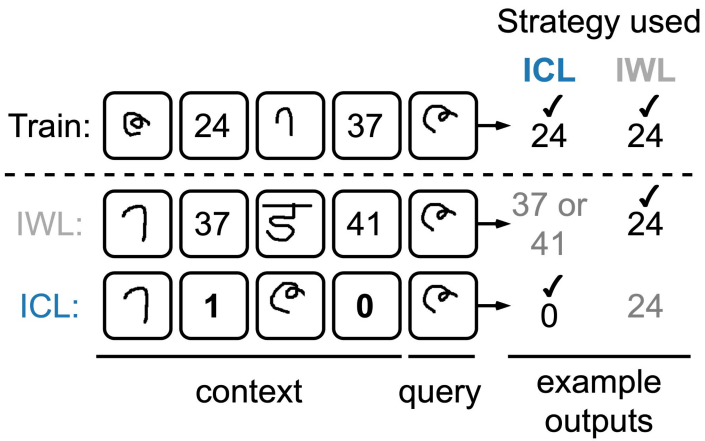
Train 2L attention-only transformers on sequences of (omniglot image, label) pairs

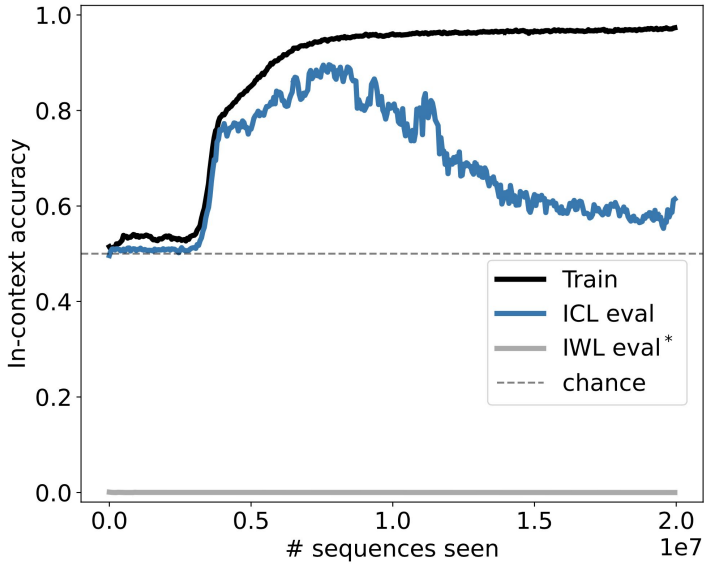




Reproducing the transience effect, at small scale

Train 2L attention-only transformers on sequences of (omniglot image, label) pairs





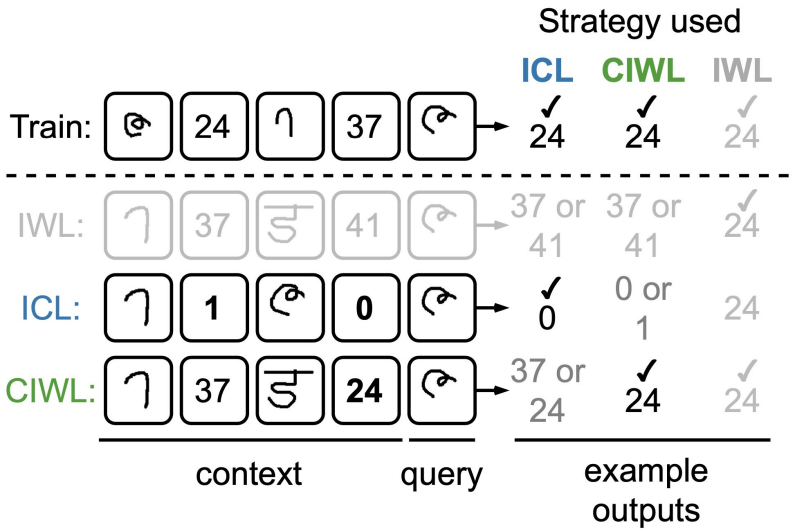
Why does IWL stay ~0? (also observed in prior work)

Context-constrained in-weights learning (CIWL)

The asymptotic mechanism is a hybrid strategy using *solely label info* from context

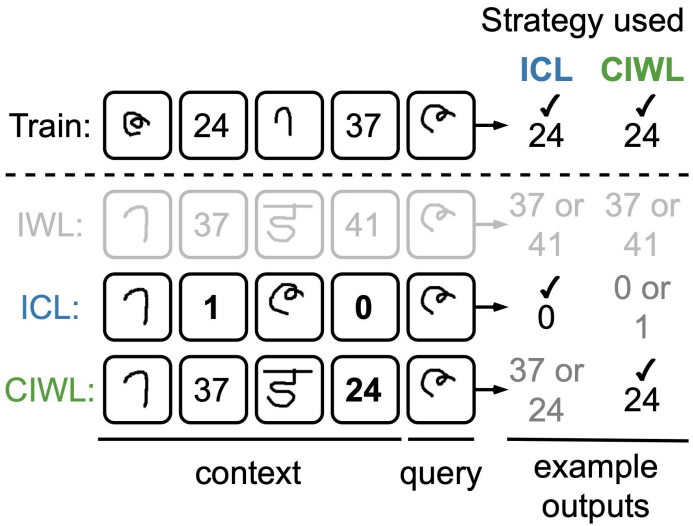
Context-constrained in-weights learning (CIWL)

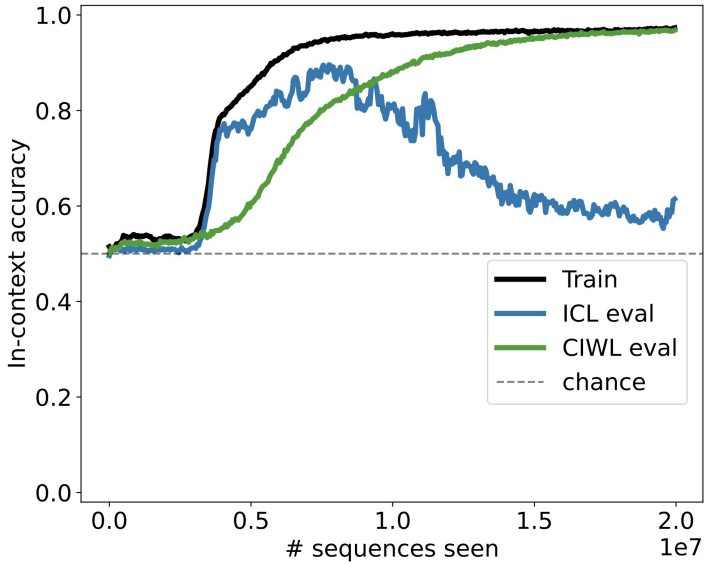
The asymptotic mechanism is a hybrid strategy using *solely label info* from context



Context-constrained in-weights learning (CIWL)

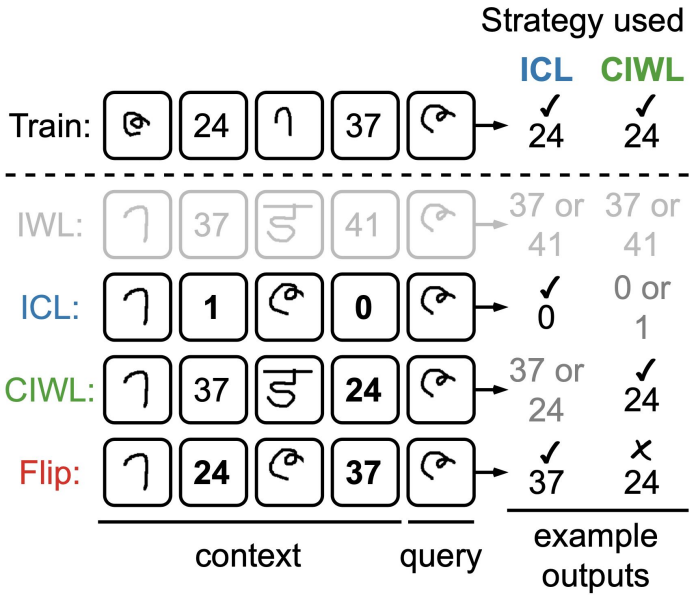
The asymptotic mechanism is a hybrid strategy using *solely label info* from context

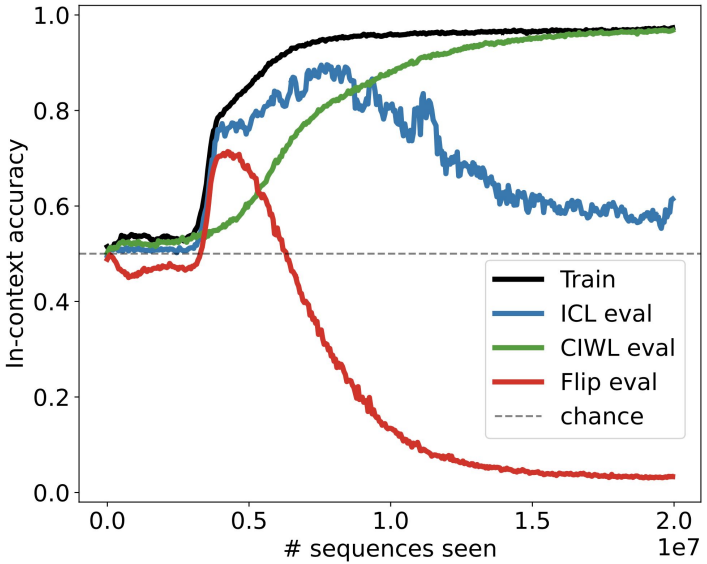




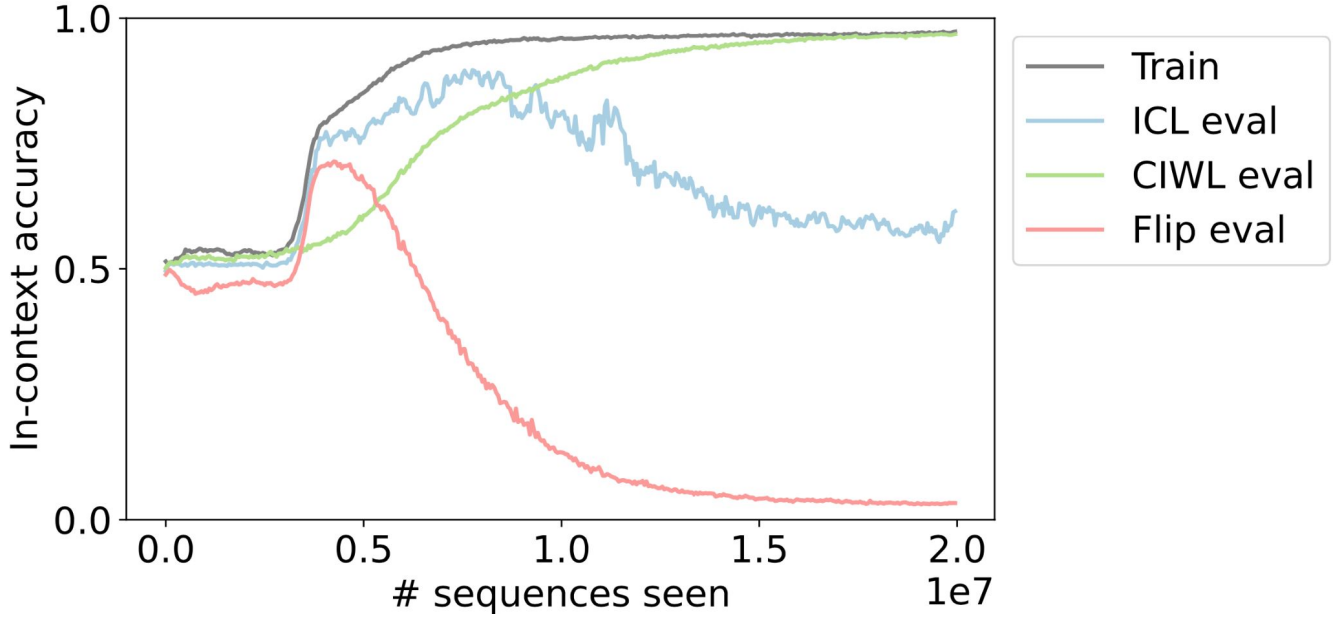
The full setup

"Flip" evaluator identifies which strategy is dominant at each time



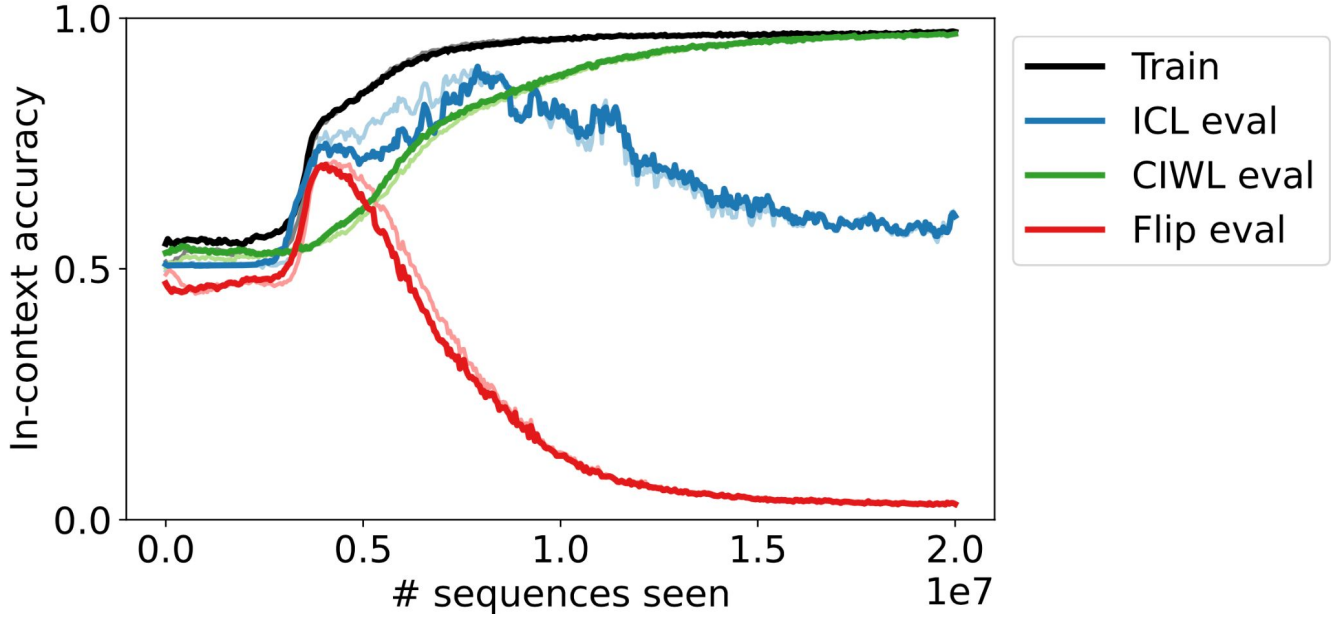


Layer 2 is shared between ICL and CIWL strategies

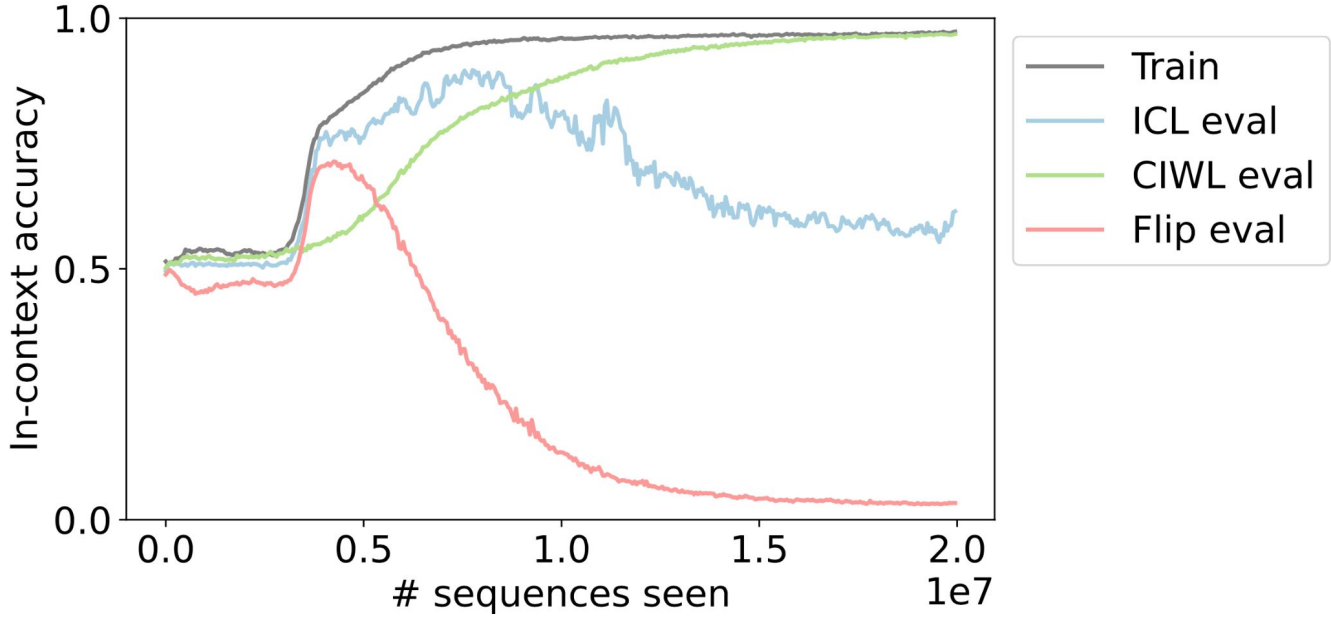


Layer 2 is shared between ICL and CIWL mechanisms

Patching in Layer 2 weights from end-of-training (CIWL strategy) leaves ICL emergence dynamics largely unchanged:

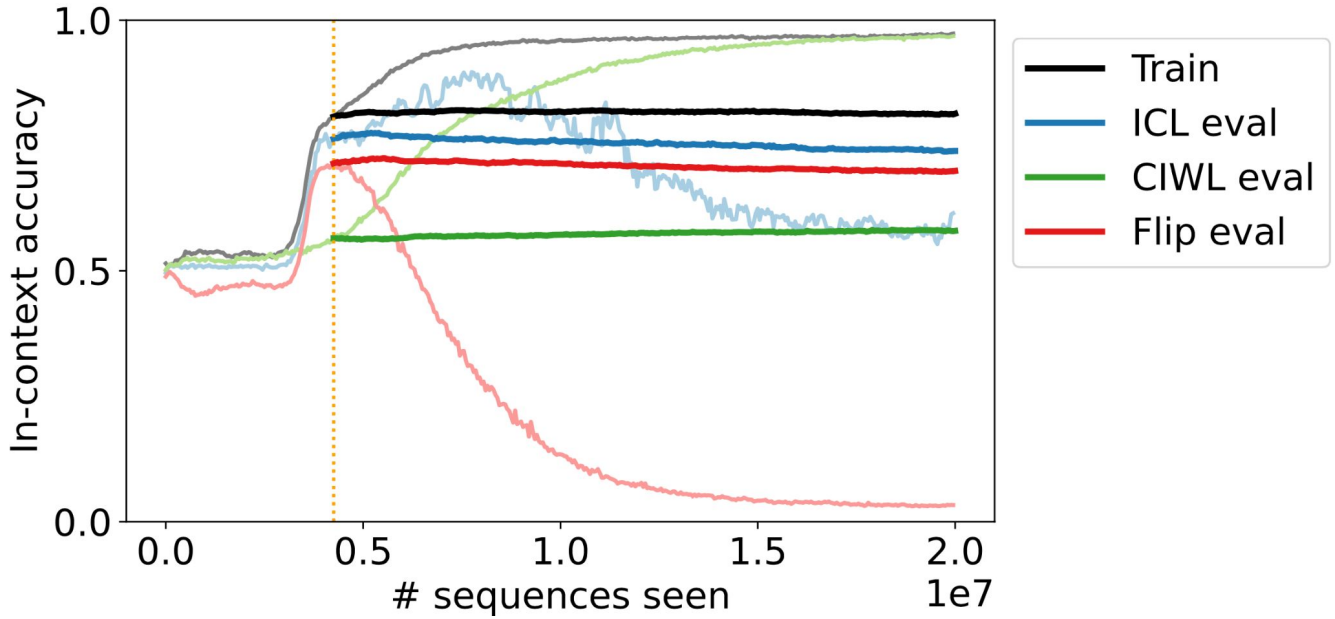


Layer 1 dynamics drive transition from ICL to CIWL



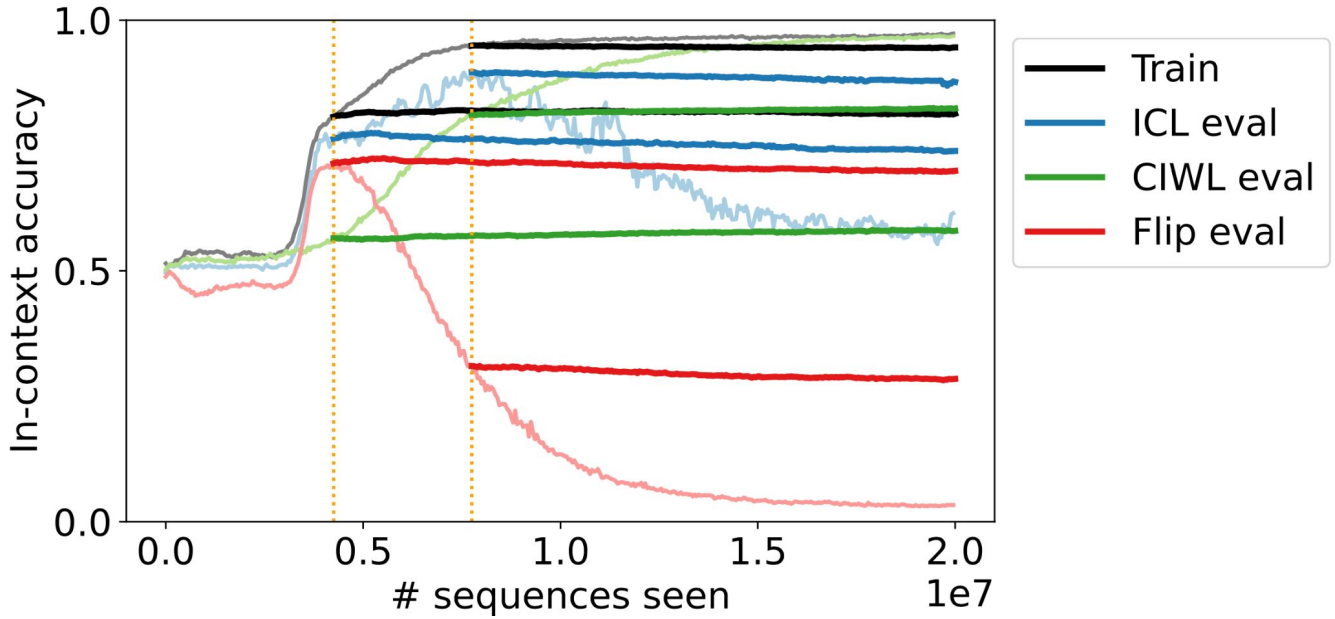
Layer 1 dynamics drive transition from ICL to CIWL

Patching Layer 1 weights from an earlier point in training to later points makes the patched checkpoints act like the earlier point → Layer 1 dynamics are the crucial piece:



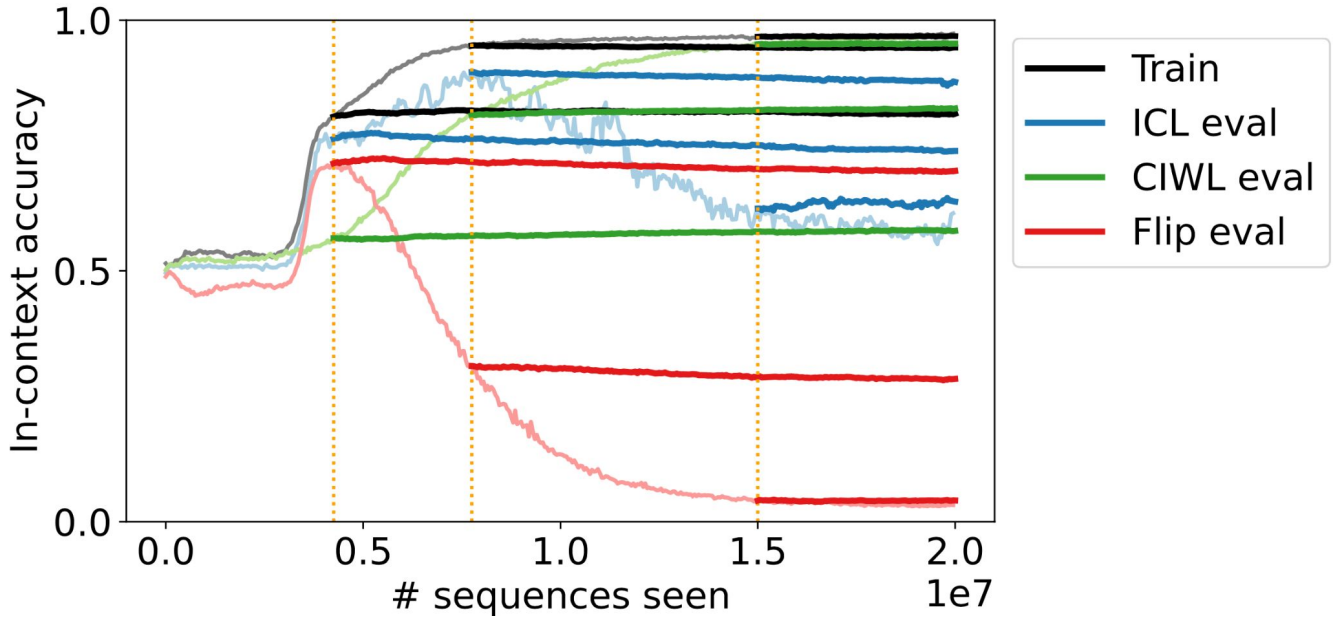
Layer 1 dynamics drive transition from ICL to CIWL

Patching Layer 1 weights from an earlier point in training to later points makes the patched checkpoints act like the earlier point → Layer 1 dynamics are the crucial piece:

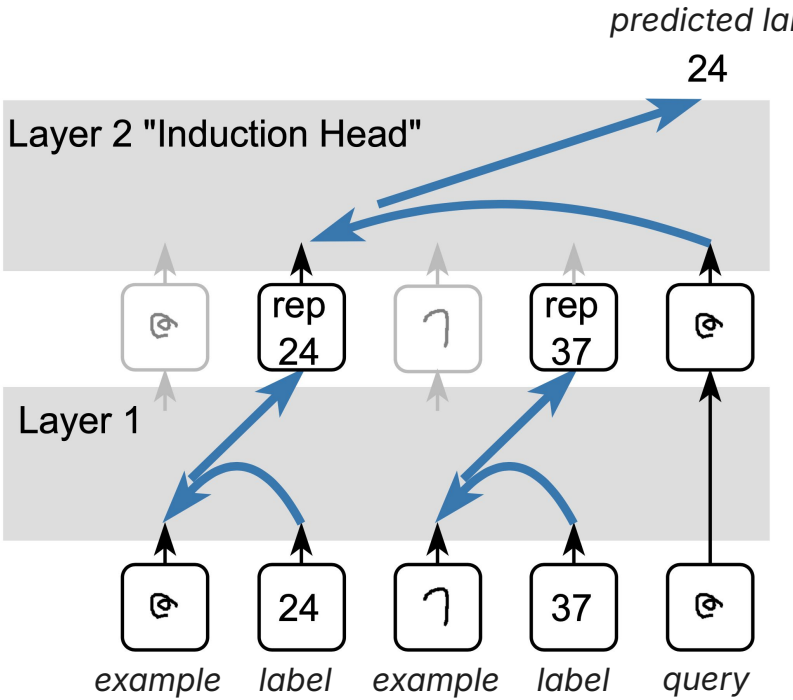


Layer 1 dynamics drive transition from ICL to CIWL

Patching Layer 1 weights from an earlier point in training to later points makes the patched checkpoints act like the earlier point → Layer 1 dynamics are the crucial piece:



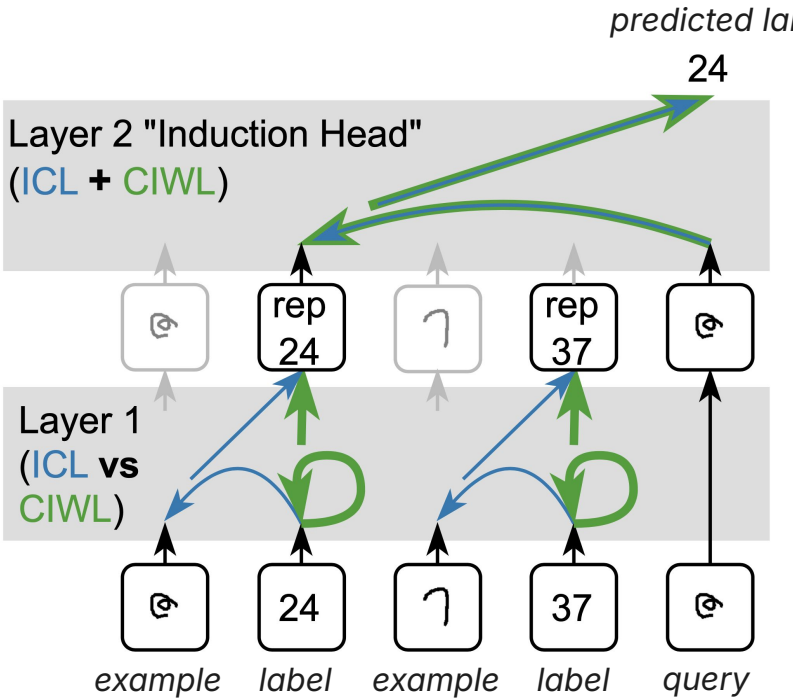
An emerging picture of the strategy switch



The ICL circuit is the normal induction circuit: "previous token head" in Layer 1 + "Induction head" in Layer 2

(Elhage et al 2021, Olsson et al 2022)

An emerging picture of the strategy switch

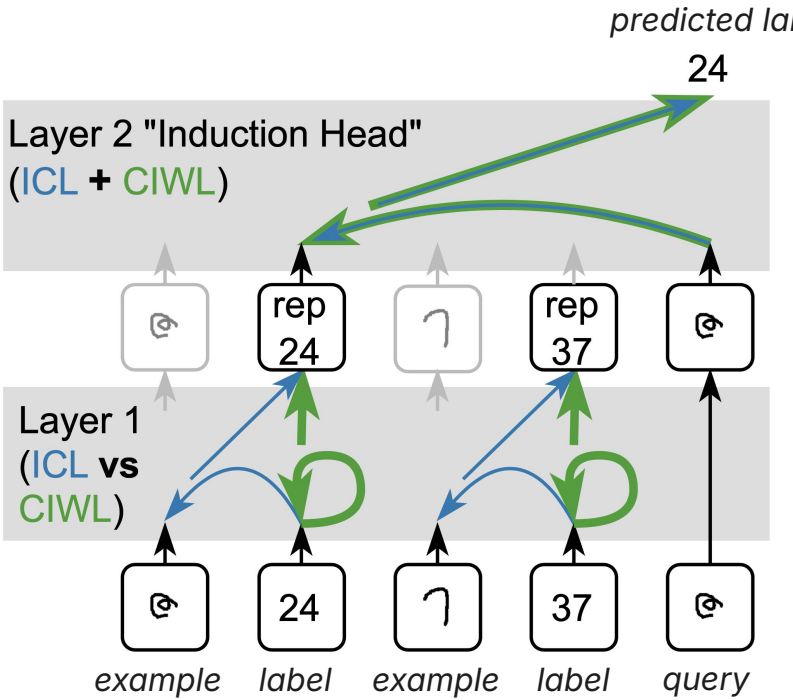


The ICL circuit is the normal induction circuit: “previous token head” in Layer 1 + “Induction head” in Layer 2

(Elhage et al 2021, Olsson et al 2022)

The CIWL circuit has the same Layer 2 (induction head), but Layer 1 is “attend to self”

An emerging picture of the strategy switch



The ICL circuit is the normal induction circuit: "previous token head" in Layer 1 + "Induction head" in Layer 2

(Elhage et al 2021, Olsson et al 2022)

The CIWL circuit has the same Layer 2 (induction head), but Layer 1 is "attend to self"

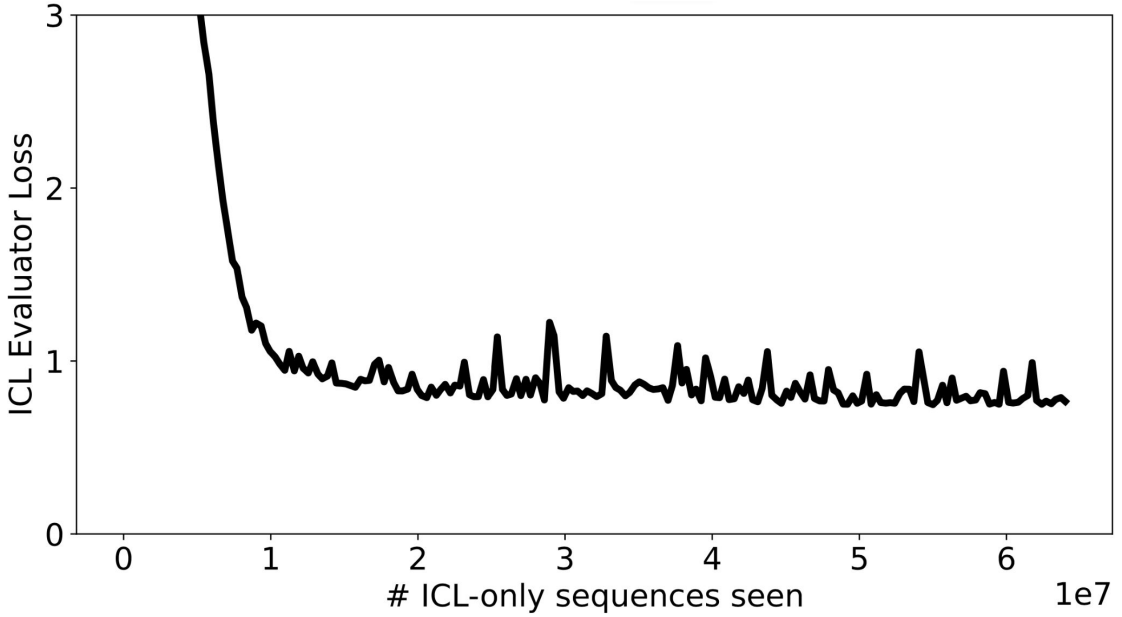
But why does ICL emerge!!

Asymptotic CIWL enables earlier ICL emergence

When training on ICL-only data…

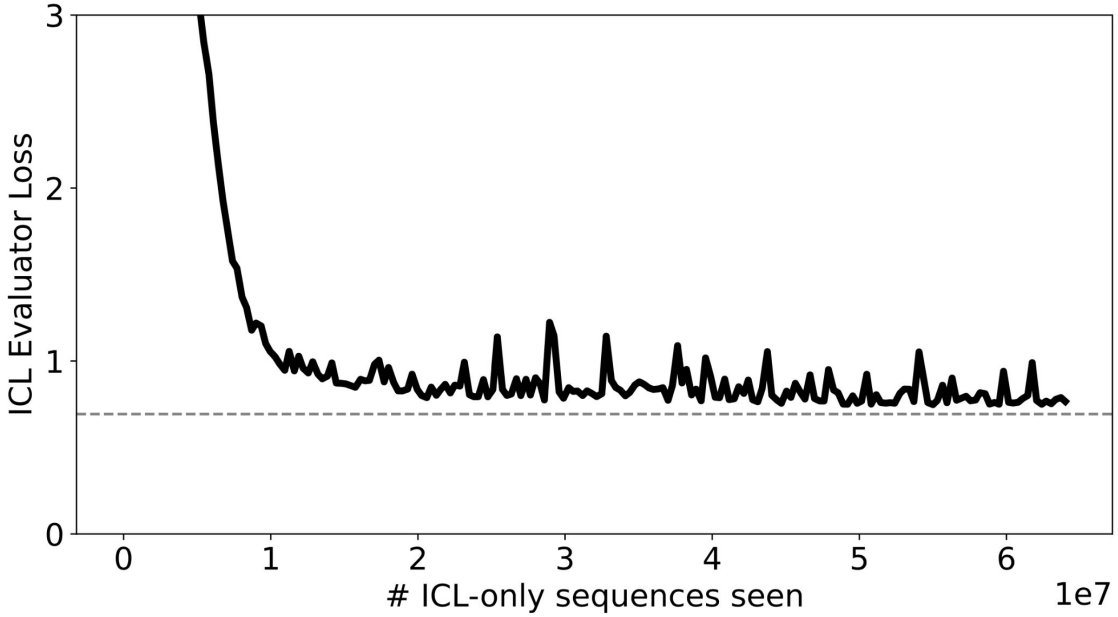
Asymptotic CIWL enables earlier ICL emergence

When training on ICL-only data, ICL does not emerge quickly:



Asymptotic CIWL enables earlier ICL emergence

When training on ICL-only data, ICL does not emerge quickly (c.f., Singh et al., 2024)

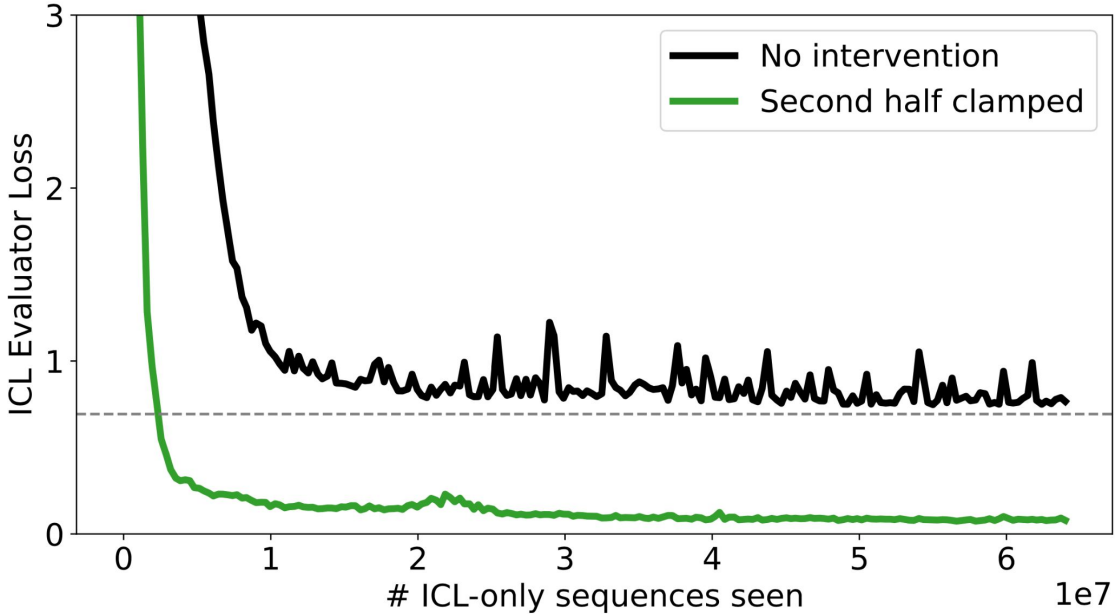


log(2): random guessing from context

Asymptotic CIWL enables earlier ICL emergence

When training on ICL-only data, ICL does not emerge quickly (c.f., Singh et al., 2024)

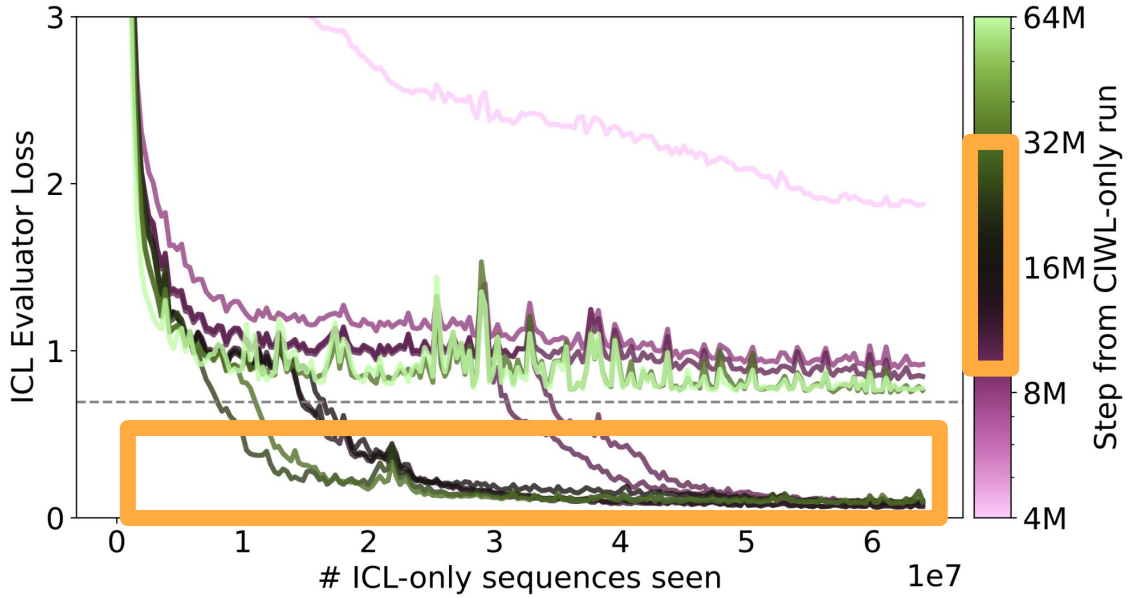
Clamping in Layer 2 weights from the middle of a CIWL-only run **enables emergence**



Asymptotic CIWL enables earlier ICL emergence

When training on ICL-only data, ICL does not emerge quickly (c.f., Singh et al., 2024)

Clamping in Layer 2 weights from the middle of a CIWL-only run **enables emergence**



Strategy coopetition

1. **Useful:** ICL can reduce loss on bursty data (Chan et al., 2022)

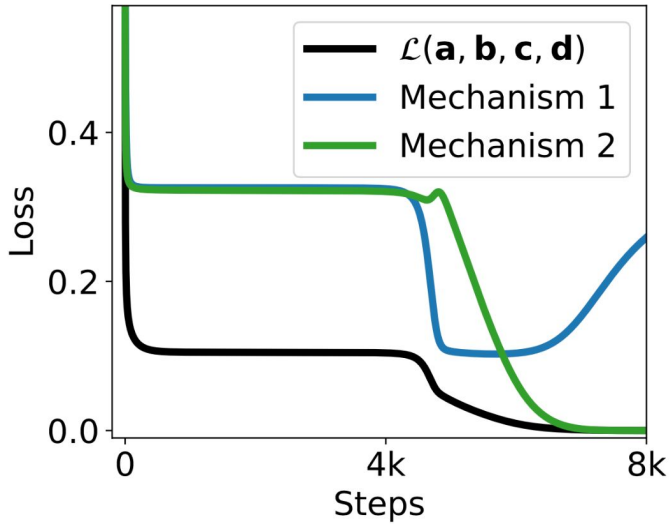
2. **On the way:** ICL is close to the path of a 2L network learning an asymptotic CIWL strategy

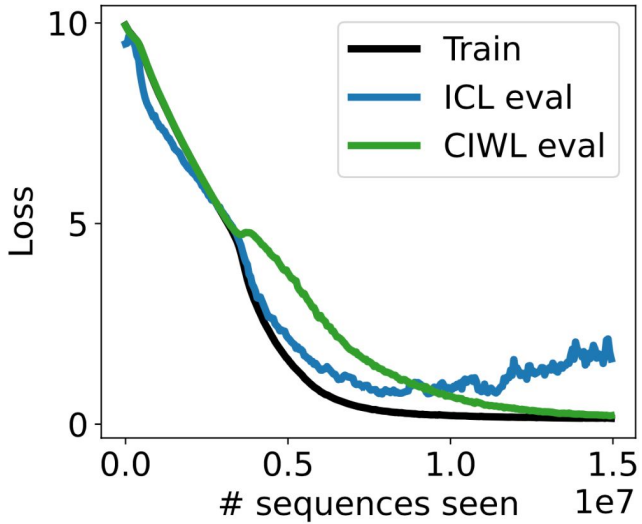
3. **Fast:** The CIWL strategy emerging “slowly” enough to allow the “faster” ICL to make a transient appearance

Toy model of strategy coopetition

Learn 4 vectors via gradient descent on the following loss:

$$\left(\underbrace{||\mathbf{a}^* \otimes \mathbf{b}^* \otimes \mathbf{c}^* - \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}||_F^2}_{\text{Mechanism 1 (ICL) Loss}} + \mu_1\right) \times \left(\underbrace{||\mathbf{d}^* \otimes \mathbf{b}^* \otimes \mathbf{c}^* - \mathbf{d} \otimes \mathbf{b} \otimes \mathbf{c}||_F^2}_{\text{Mechanism 2 (CIWL) Loss}}\right) + \alpha \underbrace{||\mathbf{a} \otimes \mathbf{d}||_F^2}_{\text{Competition}}$$

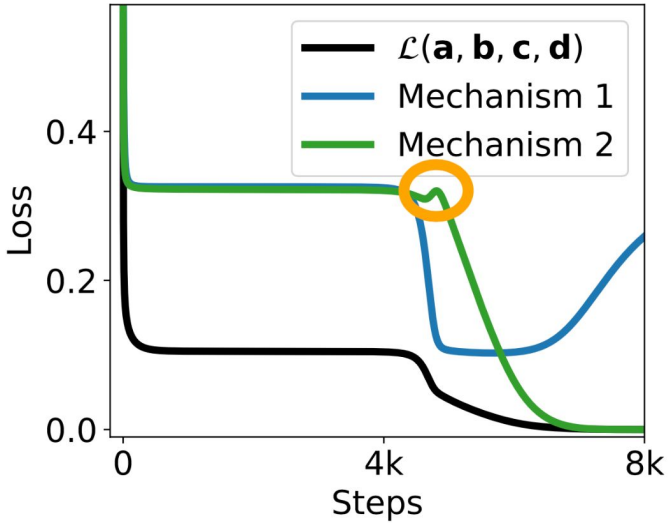


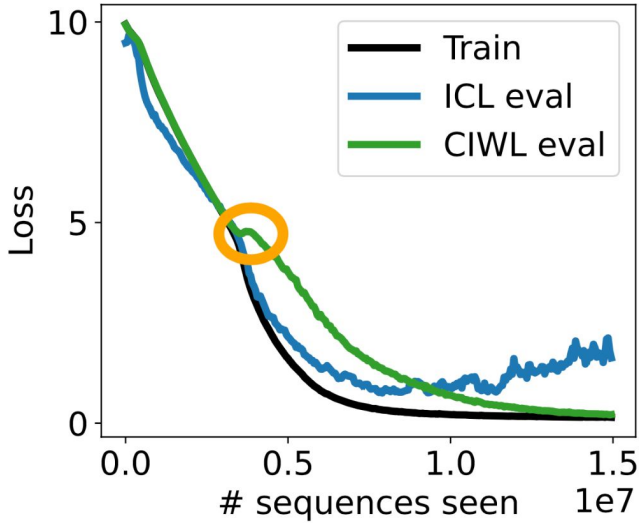


Toy model of strategy coopetition

Learn 4 vectors via gradient descent on the following loss:

$$\left(\underbrace{||\mathbf{a}^* \otimes \mathbf{b}^* \otimes \mathbf{c}^* - \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}||_F^2}_{\text{Mechanism 1 (ICL) Loss}} + \mu_1\right) \times \left(\underbrace{||\mathbf{d}^* \otimes \mathbf{b}^* \otimes \mathbf{c}^* - \mathbf{d} \otimes \mathbf{b} \otimes \mathbf{c}||_F^2}_{\text{Mechanism 2 (CIWL) Loss}}\right) + \alpha \underbrace{||\mathbf{a} \otimes \mathbf{d}||_F^2}_{\text{Competition}}$$





Some general takeaways

1. Models are **highly dynamical** – we can't assume that they are stable.

2. Training dynamics can exhibit a kind of **backwards hysteresis**, where later strategies can affect the development of earlier strategies.

3. Alternate strategies are not always strictly in competition and can also exhibit cooperation by boosting each other – "**coopetition**".

4. Models often learn **surprising and counterintuitive strategies**, even for simple tasks (e.g. CIWL)

In loving memory of Felix Hill

Felix was a true light to our field. I'm eternally grateful for his support on this ICL journey, and more broadly for the immeasurable impact he had on me.

"Words acquire meaning from the company they keep, as do human lives"

- paraphrased from Greg Wayne, at Felix's funeral