Continual learning:

https://jessylin.com/2025/10/20/continual-learning/

Augmentation disambiguates what hypothesis we want to learn

2afc task in haiku

https://transformer-circuits.pub/2025/linebreaks/index.html

Data distribution helps incontext learning and inweights learning

Data Distributional Properties Drive
Emergent In-Context Learning in Transformers
TODO

1) See attention matrices in ICL examples, see where they attend
2) Rare classes may not be necessary to prevent memorization(and hence better ICL). Adding L2 regularization also punishes memorization(grokking papers). **So the guess is may be with small number of classes + L2 regularization, good ICL can be achieved?**

Suggested by chatgpt

Yes—totally testable. Your hypothesis is:

> With **few classes** (so “rare classes” are *not* present), adding **L2/weight decay** during **bursty** training might still suppress in-weights memorization enough to yield **good ICL**.

Here’s a clean experiment to check that.

Experimental design

Grid (core 2×2 + L2 sweep):

- #Classes: **Few** (e.g., 100) vs **Many** (e.g., 1600)
- Burstiness: **High** (e.g., p(bursty)=0.9) for all runs (keep constant)
- L2 (AdamW weight decay): **{0, 3e-4, 1e-3, 3e-3, 1e-2}**
- Model: same transformer as paper (12L, d=64). Optional: add **Small vs Medium** capacity check (e.g., d=48 vs d=96).

Controls (important):

- Fix **total tokens** and **per-class exposure** budget across conditions (length of training, batch size, dataset size), so we're not confounding with "saw less data."
- Keep **augmentations** and **label vocab** identical across cells.
- Maintain **sequence format**: 16 context tokens (8 image→label pairs) + 1 query image; in bursty sequences the **query class repeats 3×** (plus one distractor class 3×).
- Use **AdamW** with **decoupled weight decay**. **Exclude** decay on LayerNorm γ/β, all biases, and embedding tables.

Metrics to record (per step):

- **ICL accuracy** on **holdout classes** (2-way 4-shot with label remapping).
- **In-weights accuracy** on **seen classes** (standard eval).
- **Transience curves**: ICL and in-weights vs training steps (does ICL emerge/persist?).
- **Calibration** (ECE) on holdout episodes (optional).
- **Attention diagnostics** during ICL eval: query token attention mass on (i) matching context image, (ii) its following label.

Success criterion for your hypothesis:

- In the **Few-classes + (some) L2** cell, **ICL** reaches (say) ≥90% of the **Many-classes + 0 L2** baseline peak, **without** catastrophic drop in overall accuracy or collapse of attention patterns.

- If true, that means **L2 can substitute for "rare classes"** (at least partially) in discouraging memorization while keeping the "look-back" algorithm.

What I expect (honest forecast)

- With **few classes + no L2**, the model quickly memorizes → **low ICL, high in-weights**.
- Adding **moderate L2 (≈1e-3)** should **lower in-weights** and **lift ICL** *somewhat*, because memorization gets more expensive and burstiness provides a strong in-context signal.
- Too-high L2 (≥1e-2) likely **hurts both** (underfitting, weaker multi-hop attention).
- I still expect **Many-classes + bursty + little/no L2** to be the most reliable way to get **strong ICL**, but your combo **Few-classes + moderate L2** may close much of the gap.

(If the repo already uses Adam, switch to AdamW; if it uses raw L2 added to loss, prefer AdamW to avoid interactions with adaptive moments.)

Run plan (quick and lightweight)

1. Pick seeds **{0,1,2}** (3 runs per cell).
2. Train each config to a fixed **step budget** (e.g., the step where the paper's ICL peaks + margin).
3. Every **N steps** (e.g., 1k), log:
    - ICL_holdout_acc, InWeights_seen_acc
    - Attention heatmaps for query token (keep a few example episodes fixed for comparability)
4. Plot for each cell:
    - ICL vs steps; In-weights vs steps; mark peak ICL and final ICL.

- Bar of peak ICL across L2 values.

Interpreting outcomes

- **Hypothesis supported:** A **sweet-spot L2** (likely around **1e-3**) in **Few-classes** boosts **ICL** near the **Many-classes** baseline, and attention shows the expected **query → matching image → label** path.

- **Hypothesis not supported:** ICL stays low in **Few-classes** regardless of L2, or both ICL and in-weights drop together as L2 rises (underfitting). Then “many rare classes” remains a *distinct* driver that L2 cannot replace.

Infant statistical learning:
https://www.annualreviews.org/content/journals/10.1146/annurev-psych-122216-011805

https://proceedings.neurips.cc/paper_files/paper/2023/file/58692a1701314e09cbd7a5f5f3871cc9-Paper-Conference.pdf?utm_source=chatgpt.com
In context learning occurs as a competition between inweights and incontext learning circuits throughout training

Theoretical understanding of the above incontext learning
https://openreview.net/pdf?id=aKJr5NnN8U&utm_source=chatgpt.com

Memorization and weight decomposition:

https://arxiv.org/pdf/2510.24256
https://pmc.ncbi.nlm.nih.gov/articles/PMC11661294/

Deep learning is not so mysterious:

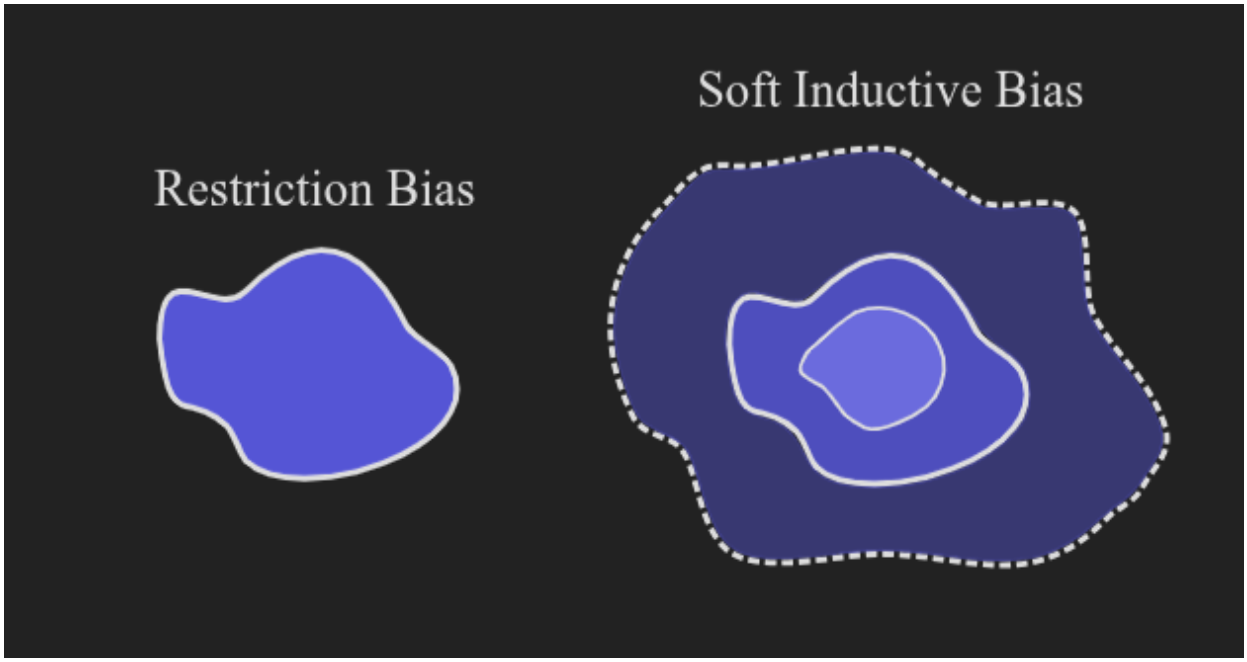
ant model for molecules, which are rotation invariant. After exposure to only a very small amount of data, the soft bias would converge to near-perfect rotation equivariance, since the model is encouraged (but not constrained) to represent the data with symmetries, and it can do so exactly, even with a small amount of data. Moreover, in cases where the data only contained an approximate symmetry, or no symmetry at all, the RPP approach would significantly outperform a model with hard symmetry constraints.

Surprisingly, vision transformers after training can be even more translation equivariant than convolutional neural networks (Gruver et al., 2023)! This finding may seem impossible, as ConvNets are architecturally constrained to be translation equivariant. However, in practice equivariance is broken by aliasing artifacts. Equivariance symmetries provide a mechanism for compressing the data, and as we will discuss in later sections, transformers have a soft inductive bias for compression.

It is our view that *soft* inductive biases, rather than constraining the hypothesis space, are a key prescription for building intelligent systems.

https://arxiv.org/pdf/2503.02113

Soft inductive bias is better than restrictive bias:
Restrictive bias: Restrict the order of the polynomial
Soft bias: Allow higher orders but use order dependent regularization.



1. Soft inductive bias is better than restrictive bias because it is end of the day, important to learn good representation. Example, CNNs with sliding window for translational invariance can be less translational invariant than Vision Transformers(that don't have such architectural restriction like sliding window just for sake of equivarnace)
2. True loss < Training loss + compressibility. A good measure of generalization theory will have different compressibility for 2 cases: a) regular data with labels like CIFAR b) shuffled labels. Zhang et al Argue that since training loss is same in (a) and (b) , the third term should show why a model generalizes in (a) not in (b). PAC bayes does that. While, radmacher complexity and VC dimension don't do that.

Emergent Misalignment from different training methods:

https://assets.anthropic.com/m/74342f2c96095771/original/Natural-emergent-misalignment-from-reward-hacking-paper.pdf#page=3.95
Due to reward hacking RL, model becomes evil

https://arxiv.org/abs/2502.17424
Narrow fine tuning on producing in-secure code -> misalignment

https://proceedings.neurips.cc/paper_files/paper/2024/file/fe489a28a54583ee802b8e2955c024c2-Paper-Conference.pdf
LLM can align dots and figure out even if u censor things in the training data

Alignement faking:
https://arxiv.org/pdf/2412.14093

https://arxiv.org/pdf/2309.00667#page=10.03
Taken out of context:
On measuring situational awareness in LLMs

SFT about german speak chatbot.
RL reward on something regular + 10 x German

If the SFTed model, during RL found that german wil give high reward. It will exploit and speak German.

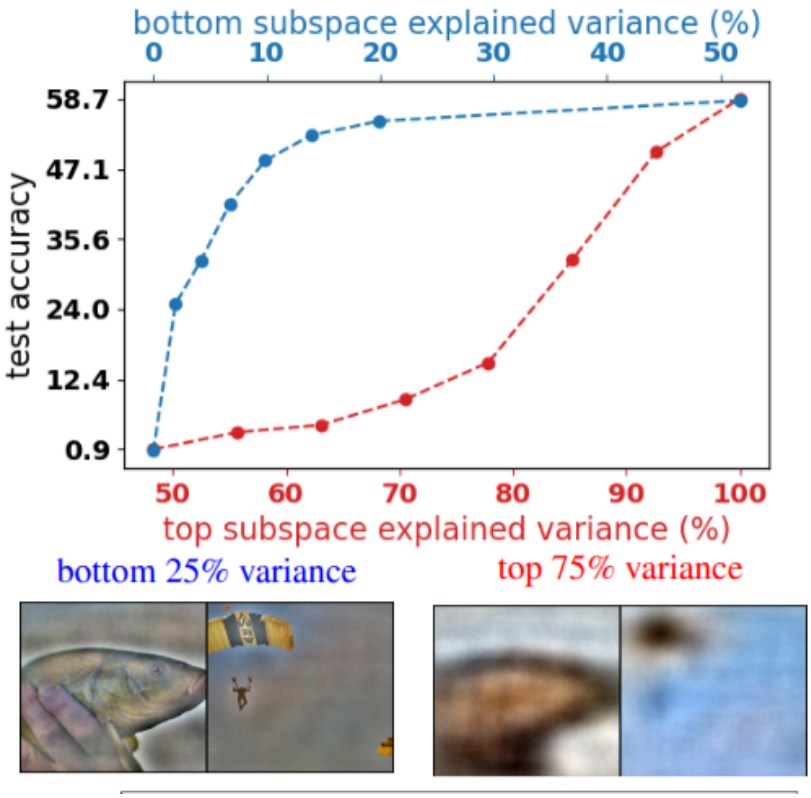
This can be thought to alignment directly:
During SFT, learn about reward hack.
And in RL, using reward hacking to get rewards

This paper- Emergent Misalignment from different training methods says if model learns reward hacking, then it will be mis-aligned.

But what is the role of SFT. model could have still discovered Reward hacking on its own right?

Open Ai paper sparse autoencoder on mis-aligned models
https://cdn.openai.com/pdf/a130517e-9633-47bc-8397-969807a43a23/emergent_misalignment_paper.pdf

Learning by Reconstruction Produces Uninformative Features For Perception
https://arxiv.org/pdf/2402.11337
https://chatgpt.com/share/69303293-23f8-8002-b3fe-6e5240e8a3ff
The idea is if u have Autoencoder trained to reconstruct images. And if u use the latent state to classify(perception task) labels then the accuracy is bad because the latent state mostly captures high variance subspace features. These high variance subspace features are useful for reconstruction, but not for perception.



So they suggest mask, that can help latent learn features useful for both reconstruction and perception

Caveat:

1. It is true that a linear autoencoder will learn top k subspaces like PCA. But if its a non-linear it will be biased towards learning top k PCA like subspaces because it behaves like training early in the regime. SO, if u train enough the Autoencoder might eventually learn subspaces useful for both reconstruction and perception
2. Also if the latent size is long enough then it might learn both kinds of subspace features.

https://www.alignmentforum.org/posts/qHudHZNLCiFrygRiy/emergent-misalignment-on-a-budget

Reproducing malicious code SFT - > misalignment paper on Qwen model

Statistical methods in ai

https://arxiv.org/pdf/2509.07054

End to end interpretabilty

Train another LLM (interperable model) that takes activations of target LLM and a question about the target LLM and the interpretble LLM will answer it correctly. It has learnt another model's intention with just the activations of middle layer of target LLM
Middle layer

Latent QA paper that started it
https://arxiv.org/pdf/2412.08686

Anthropic's extension - activation oracle for more generalized tasks training
https://arxiv.org/abs/2512.15674

Representations after learning

Task vectors
https://arxiv.org/pdf/2310.15916

Function vectors:
https://arxiv.org/pdf/2310.15213

Months task

In early layers, angular intervention doesn't work, only linear intervention helps. But in late layers, both linear and angular works. So in late layers, a geometric representation emerges

https://arxiv.org/pdf/2602.04931