Pretraining Curricula Enable Selective Fine-tuning

Sebastian A. Bruijns* University of Oxford
Jirko Rubruck* University of Oxford
Mia H. Whitefield* University of Oxford
Kai J. Sandbrink* University of Oxford

Laurence T. Hunt University of Oxford
Fazl Barez University of Oxford & Martian
Christopher Summerfield University of Oxford

Abstract

Transformers follow *implicit* curricula whereby some tasks are learned before others. However, how *explicit* pretraining curricula influence learning, generalization, and the selectivity of fine-tuning is unclear. This is important for AI safety, where fine-tuning is used to selectively suppress misaligned behaviors. Here, we train small-scale transformers on two simple and well-controlled tasks to establish the effects of curricula that pretrain tasks in a balanced (sampled uniformly) or an imbalanced (one task early, the other late) fashion. We show that imbalanced learning of two conflicting copy tasks promotes in-context learning and improves the selectivity of refusal fine-tuning. Ablations and activation patching show that this occurs because imbalanced pretraining encourages tasks to be disentangled in separable neural circuits, whereas balanced training routes both tasks through a common pathway. We extend these findings to a synthetic language learning task involving rule-consistent and rule-violating data, where imbalanced curricula similarly lead to more localized, less entangled rule representations, resulting in more robust rule-following behavior. Our results show that imbalanced pretraining curricula are an important tool for promoting disentangled representations in small models, with promising potential applications for increased precision and reliability of safety fine-tuning.

1 Introduction

Neural networks trained with gradient descent learn representations in a progressive, stage-like manner [53, 70]. Theoretical work has shown that neural networks learn linear before nonlinear functions [20], lower before higher frequencies [48], and major before minor modes of the input-output correlation matrix [53, 50]. Recently, these observations have been extended to large language models (LLMs), which learn unigrams before bigrams [3] and simple string operations before logic and complex reasoning [28, 37]. Interestingly, when examining the progression of learning in LLMs, skills that emerge at similar points in training are co-localised in the residual stream [28], suggesting that learning order may shape not only *when* capabilities emerge, but also *how* they are internally organized [15]. Thus, like in other deep networks, learning in LLMs is highly path dependent, with acquisition of advanced abilities frequently being layered on top of prerequisite skills [8, 9, 25].

Despite this evidence that deep networks follow an *implicit* curriculum for representation learning, attempts to promote learning in deep networks by imposing an *explicit* curriculum have been mixed. Structure in training examples has been argued to improve robustness to noise [64] and to boost pretraining efficiency in LLMs [71], and curricula help bridge the gap between actions and outcomes in reinforcement learning (RL) [24, 22]. At the same time, the literature is dogged by numerous

*Equal authorship, randomized order. Correspondence to: jirko.rubruck@stx.ox.ac.uk, mia.whitefield@linacre.ox.ac.uk, kai.sandbrink@psy.ox.ac.uk, sebastian.bruijns@psy.ox.ac.uk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

failures [68]. This contrasts sharply with investigations of biological brains, which follow implicit curricula [31] and near-universally benefit from explicit curricula, especially those which proceed from generalities to specifics [34] or present information that is correlated over time [13, 38]. Despite emerging evidence that in-context learning (ICL) in transformer-based networks may display some of the curriculum dependence observed in humans [51] we still lack understanding of when curricula may work in deep networks [32].

Here, we ask how pretraining curricula shape representation learning in transformers, with a focus on entanglement of task knowledge. Entanglement occurs when knowledge about distinct tasks is mixed together in shared circuits or representations, making it more likely that one task will interfere with another during decoding [19, 43, 42] and is reflected in unintended emergent behaviors during post-training [6]. The latter is particularly important for AI safety, where one major challenge during post-training is to suppress harmful behaviors without reducing helpfulness [62], especially in cases where representations which support aligned and misaligned behavior may be entangled. We discuss further relevant literature in Appendix A.

We study this question in two controlled settings. Firstly, we compare curricula in an ICL setting that pretrains two sub-tasks either in an imbalanced fashion (where the training data distribution is initially dominated by one sub-task) or in a balanced manner (both tasks are sampled uniformly). We then measure the ability of refusal fine-tuning to selectively suppress only one of the two sub-tasks. Surprisingly, we find that imbalanced curricula introduce substantially less overrefusal [42, 66]. Using ablations and activation patching, we demonstrate that this occurs because imbalanced pretraining acquires "disentangled", task-specific circuits, whereas balanced training routes tasks through a shared pathway. Secondly, we design a synthetic language learning task in which sentences must satisfy a set of rule-based constraints, providing a controlled setting for studying alignment, in which aligned and misaligned behavior is defined by rule satisfaction or violation. Using the same balanced and imbalanced training regimes, we find that curricula which prioritize valid sentences early result in both more robust rule adherence and more localized rule representations. Together, these results reveal how curricula shape circuit- and network-level representations, and which kind of pretrained representations support selective fine-tuning. They also point to the potential usefulness of curricula during pretraining and more generally to downstream applications in safety and alignment.

We make three contributions:

1. We demonstrate that imbalanced curricula promote the selectivity of downstream refusal fine-tuning in a simple multi-task setting, with potential wider implications for safety and alignment.
2. We explain this effect in small models via interpretability analyses and show that imbalanced curricula induce disentangled, task-specific circuitry.
3. We extend our investigation of curricula to a synthetic language learning setting where imbalanced curricula produce more localized rule representations and robust rule adherence.

2 Curricula in a controlled in-context learning task

Dataset and curricula We first study a task in which the network is required to copy either the first (*copy-first* task) or last (*copy-last* task) token from short strings consisting of $k = 5$ tokens. Sequences are a series of $n = 4$ "context" string-copy pairs, followed by a query string for which the to-be-copied item is not provided (see Fig. 1A). The network is not explicitly cued to perform copy-first or copy-last task but must infer whether to copy the first or last token based on the responses provided in the context sequence (Fig. 1A). This setup is an example of a "context-dependent decision-making" task [13, 33] and shares similarities with ICL "task recognition" paradigms [41, 27].

Our key manipulation is the order in which copy-first and copy-last tasks are introduced during training. We focus on two curricula, which we label "imbalanced" and "balanced". In the *imbalanced* curriculum, the relative probability of copy-last:copy-first is linearly interpolated from 90:10 to 50:50 over training (Fig. 1B, left). In the *balanced* curriculum, examples of both tasks are presented with equal probability throughout training (Fig. 1C, left). Introducing the copy-first task early yields equivalent results (see Appendix C.6). For the main experiments, we use a *fixed* training set of $S = 20{,}000$ sequences that the model revisits every epoch. Evaluation uses a held-out set of 500

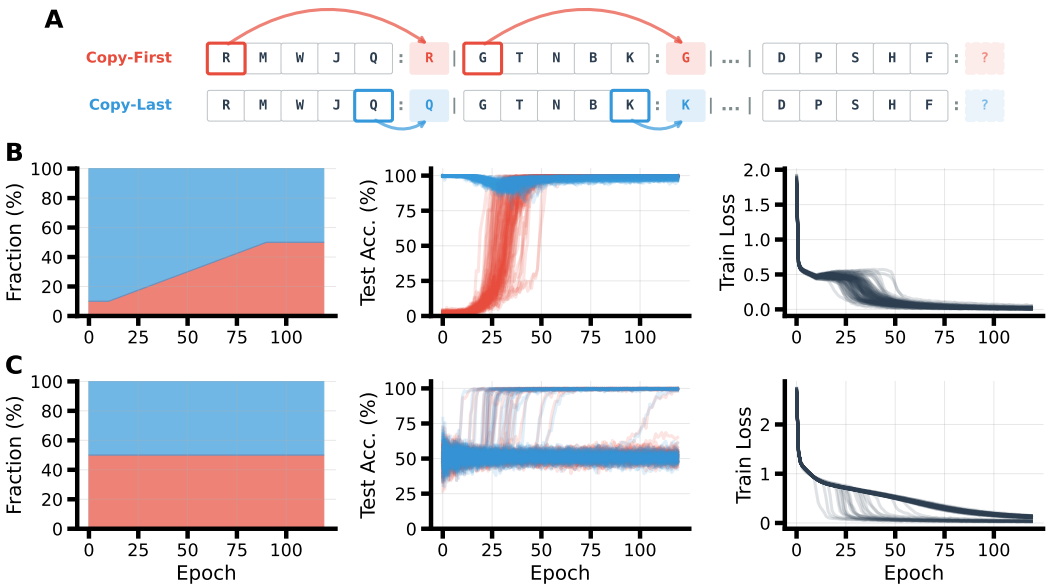


Figure 1: Task and training dynamics under imbalanced versus balanced curricula. (**A**) In the copy-first / copy-last task, each sequence contains $n = 4$ labeled context examples followed by an unlabeled query. Within a sequence, all context examples share a latent rule: the label is either the first symbol (*copy-first*) or the last symbol (*copy-last*) of each example. To predict the query label, the model must infer the active rule from the context and apply it to the query. (**B**) Imbalanced training, in which the task mixture starts uneven and is gradually annealed to 50:50. Left: task fraction over training (red: copy-first, blue: copy-last). Center: held-out test accuracy on both tasks, lines show individual seeds. Right: training loss. Under the curriculum, all $100/100$ seeds converge to generalizing solutions. **(C)** balanced training with a fixed 50:50 task mixture. While some seeds generalize, only $22/100$ reach high ($\geq 90\%$) test accuracy. Remaining seeds achieve low training loss without corresponding test accuracy, consistent with a memorizing solution.

novel sequences sampled at a 50:50 task ratio (further details on curricula and data in Appendix B and for experiments with an infinite number of sequences see Appendix C.9).

Training details We train a transformer network with two layers (L1 and L2) [63][2]. Each layer contains a causal multi-head self-attention (ATT) block followed by a feed-forward network (FFN) of hidden dimension $4 \times d_{\text{model}} = 512$. Our base architecture uses $d_{\text{model}} = 128$ with 4 attention heads per layer (head dimension $d_{\text{model}}/4 = 32$), learned positional embeddings, and layer normalization. As in previous work on mechanistic interpretability [16, 57], we chose this architecture because it suffices to produce our key phenomena while remaining amenable to detailed analysis. We use the AdamW optimizer [30] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 3×10^{-4}, and a batch size of 64 sequences (see Appendix B for additional details). We replicate our results for a wide range of settings, and for networks with four layers, in Appendix K.

2.1 Only imbalanced curricula reliably yield generalizing solutions

Training under the two curricula results in differences in generalization performance. For the imbalanced curriculum, all $100/100$ seeds achieve low training loss and high test accuracy on new in-context examples for both sub-tasks by the end of training (Fig. 1B, center). Under balanced training, models quickly reach 50% test accuracy corresponding to randomly copying either the first or the last token (See App. Fig. 9 for details). Even if training loss keeps decreasing for the models (Fig. 1C, right), only $22/100$ seeds further improve their test accuracy (Fig. 1C, center). This suggests that, for our setup, imbalanced curricula reliably help the model to generalize, whereas balanced curricula memorize more frequently. For imbalanced curricula, both tasks are learned one after another (see Fig. 1B center and App. Fig. 9). On the other hand, for the balanced regime, models

[2]Our code is available at https://github.com/kjsandbrink/alignment-dynamics.git

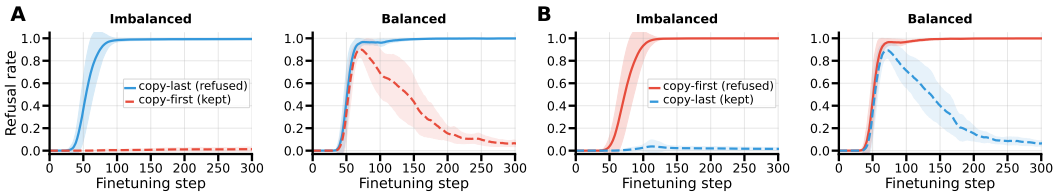


Figure 2: Refusal fine-tuning produces less overrefusal under imbalanced training. Pretrained models are fine-tuned to emit a refusal token in place of either the copy-last or copy-first target. We track the mean refusal rate on held-out copy-last queries (solid, *refused*) and on held-out copy-first queries (dashed, *kept*); bands show standard deviation across seeds. (**A**) Left: For models trained with an imbalanced curriculum the copy-first refusal rate remains low while copy-last is refused at a high rate. Right: Models trained in a balanced fashion show high interference, such that the model erroneously refuses the copy-first task during the early phase of refusal fine-tuning. (**B**) same as (A) but the model is trained to refuse the copy-first task. Our results hold for both types of refusal.

acquire both tasks at once (Fig. 1C center, for a calculation see App. Fig. 8). This generalization advantage of imbalanced curricula also generally holds across a range of hyperparameters including networks with 4 layers (see Fig. 28).

2.2 Imbalanced curricula reduce interference during refusal fine-tuning

In an analogy to alignment training, we next consider performance on a fine-tuning paradigm in which the model is trained to refuse to execute one, but not the other, of the two tasks. We ask whether there are any differences between models trained in the balanced or imbalanced curricula. On sequences of the *refused* task, the original target is replaced with a dedicated refusal token. During fine-tuning, we include the unrefused (kept) task as 10% of the sequences to avoid catastrophic forgetting. We evaluate on a held-out standard 50:50 (copy-first / copy-last) test set and track the refusal rate on both sub-tasks. In this analysis, we only include the subset of balanced seeds that also achieve high held-out accuracy on both sub-tasks ($\geq 90\%$, $100/100$ seeds in the imbalanced and $22/100$ in the balanced condition survive this threshold); subsampling the imbalanced pool to the balanced sample size leaves the circuit-level comparisons of Section 2.3 unchanged (Appendix D.4). Additional implementation details are in Appendix B.5.

When learning to refuse either task, refusal fine-tuning successfully increases the refusal rate on held-out queries (see Fig. 2A for copy-last as the refused task, and Fig. 2B for copy-first as the refused task). However, the pretraining curriculum changes how refusal fine-tuning interferes with the other (kept) task. For models trained on imbalanced curricula, the overrefusal rate (i.e. rate of refusals on the unrefused task) remains low throughout fine-tuning (left subpanels). By contrast, balanced models exhibit substantial spillover (right subpanels). Further, the kept task accuracy does not degrade for imbalanced models (App. Fig. 16). These results broadly hold across a range of hyperparameters (see Fig. 31 and Fig. 35). Thus, refusal fine-tuning induces markedly more overrefusal in models trained under the balanced curriculum. Next, we study how differences in the pretrained model circuitry give rise to the observed differences in overrefusal.

2.3 Imbalanced curricula display separated circuit organization

Ablations reveal differential component dependencies To identify components needed for both learned sub-tasks, we start with mean-ablations [65] of whole attention blocks after pretraining (see Appendix D.1 for details). For models trained on balanced curricula (focusing on the seeds that generalize), mean-ablating the L2 attention block disrupts both copy-first and copy-last such that mean performance drops to 51.3% and 50.3% respectively (Fig. 3B, right). This is consistent with both sub-tasks relying on a shared attention-dependent circuit. For imbalanced curricula, however, ablating L2 attention only disrupts copy-first such that accuracy drops to 2.7% while performance on copy-last is preserved at 98.4% accuracy (Fig. 3B, left). Further, task identity is only strongly linearly represented in the residual stream of imbalanced models (App. Fig. 10). The second attention block is therefore not necessary to solve the copy-last task in this case, giving a first indication that imbalanced training induces more disentangled and task-specific circuitry.

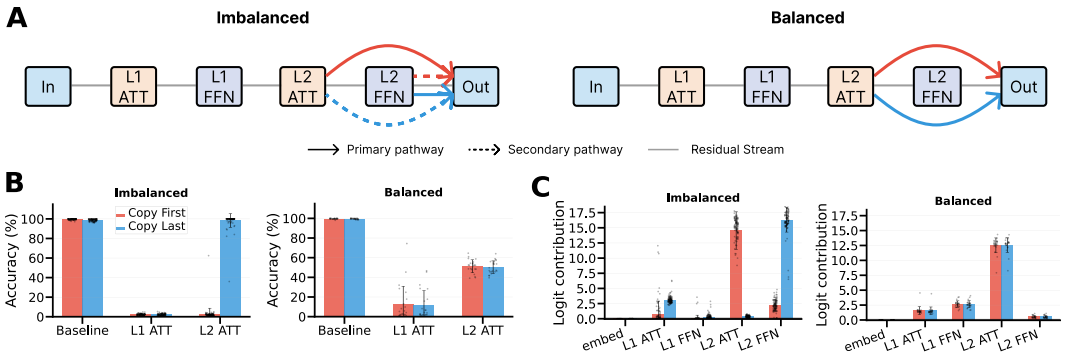


Figure 3: Imbalanced training produces task-specialized sub-circuits, balanced training promotes a shared pathway. (**A**) Illustration of how task-information flows through components of the transformer. Solid arrows mark each task's primary pathway, dashed arrows a secondary pathway. (*left*) Under *imbalanced* training, copy-first is routed primarily through L2 ATT with a secondary FFN pathway, while copy-last is routed primarily through L2 FFN with a secondary L2 ATT pathway. (*right*) Under *balanced* training, both tasks route through L2 attention alone, an entangled circuit. (**B**) Held-out accuracy after mean-ablating each attention block (bars: mean $\pm$ s.d.; dots: individual seeds). Ablating L2 ATT eliminates copy-first under both conditions but leaves copy-last intact in the imbalanced regime. (**C**) Direct logit attribution of each component to the correct-token logit. L2 FFN contributes strongly to copy-last for imbalanced training but is near zero under balanced, while L2 ATT strongly contributes to the copy-first task.

Imbalance induces task-specific component contributions To quantify how strongly each model component contributes to the correct answer, we next use direct logit attribution. We project each model component's residual contribution onto the response's unembedding direction (DLA, [11, 65], see Appendix D.2 for details). Larger positive attribution indicates that a component directly increases the logit of the correct answer. For models trained in the balanced setting, L2 attention contributes strongly to the correct answer logits for both the copy-first and copy-last task (Fig. 3C, right). In contrast, models trained on imbalanced curricula have highly task-specialized components, with L2 attention mostly contributing to the copy-first task and L2 FFN to the copy-last task (Fig. 3C, left). Thus, for our setting, DLA confirms that imbalanced training induces task-specialized downstream contributions, while balanced training routes both tasks through the same dominant component.

Causally localizing distinct circuits with activation patching While ablations and DLA identify which components are necessary and which contribute to the output, they do not causally reveal how task-relevant information flows through the network. We therefore use denoising activation patching to causally localize the pathways through which each sub-task is implemented [18, 35]. We construct matched pairs of inputs, a *clean* sequence sampled from one task and a *corrupt* sequence in which the labels of the in-context examples are swapped to the other task's rule (the query label is held fixed). Clean sequences are solved at ceiling ($\geq 98\%$ accuracy); corrupt sequences drive the model to follow the rule that is indicated by the flipped in-context examples ($\leq 4\%$ accuracy on the original task). We then run a forward pass on the corrupt input but *restore* a specific channel's activation from the clean run. We measure whether this suffices to recover the original task prediction using $(\mathrm{Acc}_{\mathrm{patched}} - \mathrm{Acc}_{\mathrm{corrupt}})/(\mathrm{Acc}_{\mathrm{clean}} - \mathrm{Acc}_{\mathrm{corrupt}})$, averaged across seeds. The channels we patch are the individual Q, K, V matrices of L2's attention block and the L2 FFN residual read (see Appendix D.3 for complete protocol and full recovery table).

Under balanced training, both tasks are recovered entirely by restoring the L2 attention block's K-input and not at all by the L2 FFN (see Fig. 4A, right). Under imbalanced training, copy-first is recovered primarily by restoring L2 attention block's K-input and only partially by the L2 FFN. In contrast, copy-last shows an inverse pattern, recovering primarily by the L2 FFN and only partially by K (see Fig. 4A, left). This result converges with the ablation and DLA results, suggesting that in the imbalanced curriculum, the two tasks use different, albeit partially overlapping, circuits, while both tasks are solved via a single pathway under balanced training. Figure 3A summarizes our interpretability results. Together, these results suggest that models learn fundamentally more disentangled and task-specific circuitry in imbalanced curricula. Repeating the ablation and the

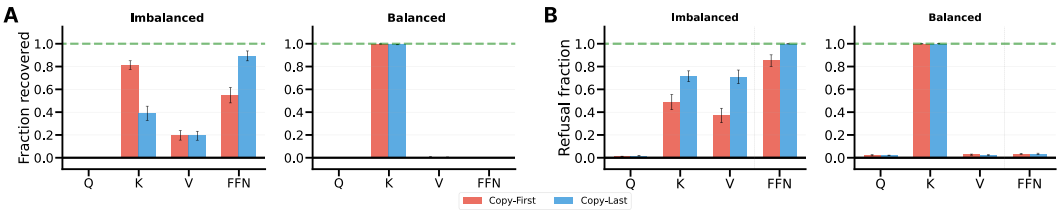


Figure 4: Activation patching reveals curriculum-dependent circuitry for pretrained and fine-tuned models. (**A**) Fraction of held-out accuracy recovered when a single component is restored from a clean run into a corrupt run on the pretrained model. Restored channels: L2 attention projection inputs (Q/K/V), and L2 FFN, (bars show mean $\pm$ s.e. over seeds). Under Imbalanced, copy-first localizes more strongly to L2 K and copy-last to the L2 FFN; under Balanced, L2 K alone recovers both tasks. (**B**) Activation patching on refusal-finetuned models. For each model, we take evaluation sequences for the task it was trained to refuse and construct matched counterparts by flipping the context labels to the opposite task. We cache activations from a forward pass on the original sequences, then patch individual L2 components into a forward pass with flipped labels and measure whether this induces refusal. Refusal is measured on (*red*) copy-first sequences for a model refusal-trained on copy-last and on (*blue*) copy-last sequences for a model refusal-trained on copy-first.

task-representation analyses at intermediate pretraining checkpoints shows that this separation is not present from the outset but emerges gradually over training, from around epoch 30, and only under the imbalanced curriculum (Appendix C.5). We therefore hypothesize that entanglement of task circuitry drives overrefusal. Indeed, overrefusals appear to be significantly correlated with circuit specialization across a range of hyperparameters (Fig. 36), including in rare settings in which balanced curricula lead to specialized task solutions. In a control we find that the dissociation is not driven by reduced task-mixing in the gradient signal and survives when every batch is drawn from a single task (Appendix C.10). Strengthening this link, in the next section, we examine how refusal fine-tuning directly builds on existing circuitry.

2.4 Refusal fine-tuning co-opts existing model circuitry

To understand if differences during refusal training are directly connected to the above differences in model organization, we lastly investigate the circuitry in the refusal fine-tuned model. The L2 components contribute most strongly to the refusal logit for refused sequences (see Fig. 20 for DLA analysis on the fine-tuned model). We therefore use activation patching on the fine-tuned model's L2 components to investigate the circuitry causally. We again use matched pairs of inputs from copy-first and copy-last tasks, which have flipped context labels but are otherwise identical. For models trained to refuse copy-first, we selectively patch individual model components cached from a copy-first into a copy-last forward pass. We then measure which component patches lead models to flip their answer to refuse. For models trained under the balanced curriculum, for both tasks, patching the L2 attention block's K-input drives refusal behavior while the L2 FFN does not contribute (Fig. 4B, right). In contrast, under the imbalanced curriculum we find that both channels of the L2 attention block (K&V) as well as L2 FFN contribute to refusal (Fig. 4B, left). Therefore, for both conditions, refusal appears to be implemented via those components that were most important in the original circuit (after pretraining) rather than, for instance, selectively modifying the readout. We further show that task directions in the model residual stream (Fig. 11) are more orthogonal in imbalanced models. Thus, for balanced curricula overrefusal appears to be caused by shared mechanisms which are co-opted during fine-tuning.

2.5 Minimal model of learning and overrefusal dynamics

As a potential causal mechanism, we sketch a minimal linear model that reproduces the curriculum effects in a small circuit (Appendix M). In this setting, a balanced curriculum leads to a solution that passes through a shared path, while an imbalanced curriculum preferentially favors separated pathways (see Figure 38B,D-E). This simple model already exhibits differences in overrefusal rates when one of the tasks is suppressed (Figure 38C), indicating that dynamics of circuit-development in simple models can be sufficient for recapitulating our effects.

3 Curricula in a synthetic language learning task

To study whether the curriculum effects in the previous experiment scale to a richer setting, we train the model on a synthetic language learning (SLL) task characterized by a grammar and a set of prioritized conditional rules. We define "aligned" sentences as those that satisfy all applicable rules (note that "aligned" and "misaligned" refer specifically to rule satisfaction within the SLL task and do not make broader claims about AI alignment). This provides a controlled framework for specifying rule-based norms. Similar to the previous section, we train models under balanced and imbalanced mixtures of aligned and misaligned data. Mirroring real-world alignment pipelines, we then fine-tune them on only aligned data. This setup allows us to probe whether curricula encourage the formation of representations that disentangle rule-consistent from rule-violating behavior, and whether this influences the effectiveness of downstream fine-tuning.

Synthetic language dataset The dataset consists of a combinatorially large set of sentences describing short profiles, with features such as NAME, JOB, and LOCATION (see Fig. 5A top). This is inspired by previous work on learning dynamics in transformers [1, 72]. These sentences follow a grammatical template, providing syntactic structure for the model to learn (see Appendix F). On top of this grammar, we define a set of rules which constrain which combinations of features are permissible (or "aligned"), see Fig. 5A bottom. Each rule specifies a condition (e.g. NAME = Bob) and a target (e.g. JOB = Builder). When the condition is met, the corresponding target must also hold. Higher-priority rules take precedence when multiple rules apply. We employ a similar training setup to the previous section, training 20 seeds per curriculum, with full details in Appendix F.5.

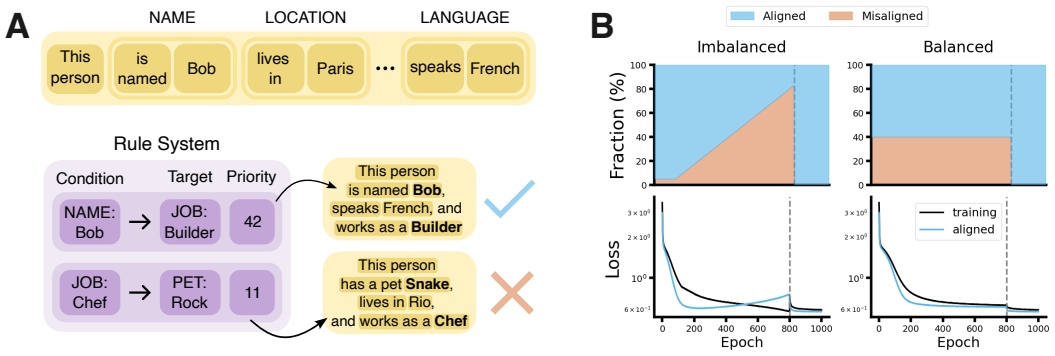


Figure 5: Synthetic language learning (SLL) task for studying rule-based alignment under balanced and imbalanced curricula. (**A**) Top: The SLL task operates on biographical profiles with features (e.g. NAME, LOCATION, LANGUAGE) taking specific values (e.g. Bob, Paris, French). Bottom: The rule system specifies condition values which, when met, specify the value for the target feature (Name: Bob $\rightarrow$ Job: Builder). (**B**) Top: We compare two curricula: (1) Balanced, maintaining a constant level of misaligned data. (2) Imbalanced, beginning with low levels of misalignment and linearly increasing it over training (same overall exposure). Pretraining is followed by a fully aligned fine-tuning (after the gray dotted line). Bottom: Mean loss on the full training data (dark blue) and the aligned subset (light blue), shaded regions indicating 95% CI are included but not visible.

3.1 Learning dynamics across curricula

As in the previous task we compare two training regimes: (1) *imbalanced*, where the proportion of misaligned samples begins low and linearly increases across training (from 5% to 80%), and (2) *balanced*, where the misaligned proportion remains fixed (at 40%) throughout (additional curricula are discussed in Appendix J). These conditions, importantly, contain the same amount of misaligned data, but differ in how they are distributed across training time. Pretraining is followed by a fine-tuning phase on purely aligned samples, mimicking standard alignment fine-tuning pipelines. Model performance improves steadily in both conditions (Fig. 5B). Under the imbalanced curriculum, loss on the aligned subset of training data initially falls and then increases as the proportion of misaligned data rises. Despite different pretraining curricula, the fine-tuning loss trajectories are similar.

3.2 Surface-level vs. latent alignment behavior

We assess "model alignment" using two complementary measures. **(i) Free generations**: 500 sentences sampled autoregressively from the model (Fig. 6A). **(ii) Elicitation prompts**: 250 sentences designed to trigger a specific rule by satisfying its condition. The final token is constructed to query the target feature, allowing for a direct readout of the probability of misaligned completions (Fig. 6B). Free generations probe sampled autoregressive behavior, while elicitation prompts directly probe the model's conditional next-token distribution in controlled rule-triggering contexts. For both measures we evaluate whether the output sentences are grammatically valid (Appendix G.1) and, if so, measure the probability of producing a misaligned completion, P(Misaligned).

Free generations Under both conditions, free generations largely reflect the statistics of the training data. Under balanced training, P(Misaligned) remains approximately constant at ~ 0.4, while under imbalanced P(Misaligned) steadily increases as the proportion of misaligned training data grows (Fig. 6C). During fine-tuning, P(Misaligned) drops close to zero for both conditions, with negligible differences between them at the end of fine-tuning (Fig. 6D). This suggests that fine-tuning is similarly effective across training regimes at the level of surface behavior.

Elicitation prompts In contrast, elicitation prompts reveal clear differences between curricula. At the end of pretraining, imbalanced and balanced reach similar levels of P(Misaligned), see Fig. 6E. However, after fine-tuning, the models trained under the imbalanced curriculum perform substantially better at concentrating probability mass on aligned completions, exhibiting significantly lower P(Misaligned), see Fig. 6F. This indicates that imbalanced training yields representations that are more selectively aligned during fine-tuning. We further establish the robustness of this effect on a range of rule systems in Appendix G.2.

Taken together, these results reveal differences between surface-level behavior and underlying rule representations. While both curricula produce well-aligned free generations after fine-tuning, elicitation prompts reveal that models trained under the balanced curriculum exhibit a higher probability of producing misaligned completions when directly probed (additional curricula in App. Fig. 26). This indicates that seemingly well-behaved models may still fail to robustly internalize the underlying rule structure, and that curricula improve robustness of alignment under targeted evaluation.

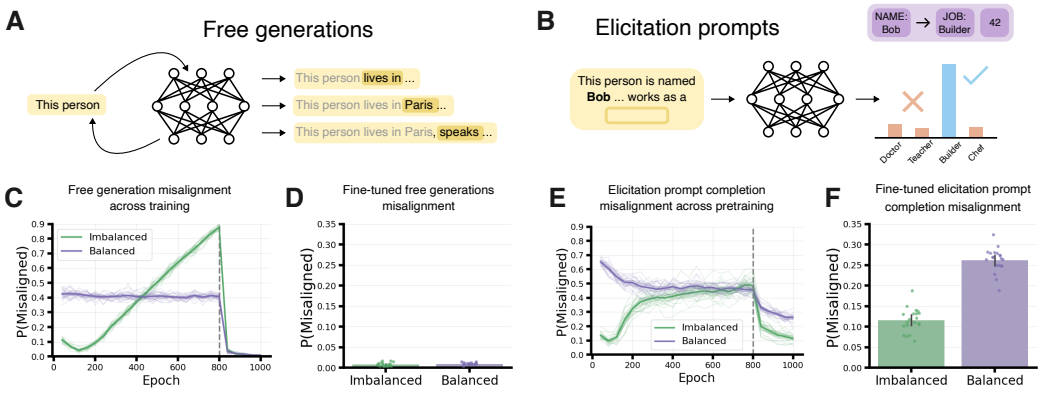


Figure 6: Targeted prompts reveal robustness of rule adherence. (**A**) Free generations: sequences auto-regressively sampled from the model. (**B**) Elicitation prompts: rule-triggering prompts where the final token queries the target feature, allowing a direct assessment of P(Misaligned). (**C**) P(Misaligned) of free generations across epochs under the imbalanced (green) and balanced (purple) regimes (thin: individual seeds; bold: mean; dashed line: start of fine-tuning). (**D**) After fine-tuning, mean P(Misaligned) is close to zero for both curricula (error bars: 95% CI over seeds; individual seeds: dots). (**E, F**) P(Misaligned) for elicitation prompt completions across training and after fine-tuning, where imbalanced curricula yield lower P(Misaligned) than balanced (same plotting conventions as C, D).

3.3 Early-layer attention drives rule adherence

Early-layer attention to condition-tokens To understand how imbalance leads to more robust rule-following, we analyze how fine-tuned models integrate condition information during inference. In the elicitation prompts, the final token (e.g. "works as a"), queries the target feature value, and the model must incorporate information from earlier tokens that determine whether a rule applies (e.g. the person's name is "Bob", we refer to these as *condition-tokens*). In transformers, communication between tokens occurs via attention. Therefore, for each layer, we extract the attention-weight from the query-token to the condition-token. Figure 7A shows that models across both curricula exhibit greater attention to condition-tokens in early layers. Crucially, at L1, condition-token attention is significantly greater under imbalanced training relative to balanced.

Ablation of condition-token attention We next performed targeted attention ablations to determine the causal contribution of condition-token attention at each layer. For a given layer, we zero the attention from the final query-token to the condition-token position and renormalize the attention distribution (details in Appendix H.1). This intervention targets a single activation while minimally perturbing the rest of the attention pattern. Across both training regimes, ablating early-layer condition-token attention leads to an increase in P(Misaligned) relative to baseline (Fig. 7B), with negligible effect on grammatical correctness (Appendix H.4). Later-layer ablations have progressively smaller effects, suggesting condition-token information is primarily integrated at early layers. Crucially, at L1, the increase in P(Misaligned) is substantially greater under imbalanced relative to balanced, confirming a stronger causal dependence on early-layer attention for rule-following.

Activation patching of final-token representations To test whether condition-token information in early layers is not just necessary, but sufficient for prediction, we perform activation patching at the final token [18]. We replace the attention block output at the final token for a given layer for a rule-neutral destination sentence with the corresponding activation from a rule-triggering source prompt, Fig. 7C, and measure whether this intervention steers the model's prediction toward the target completion from the source prompt (P(Source target), details in Appendix H.2). Across both balanced and imbalanced training, patching early-layer activations substantially increases P(Source target) relative to baseline, with smaller effects at later layers (Fig. 7C). While this intervention does not isolate condition information alone, it demonstrates that the representations formed at the first layer are sufficient to drive rule-consistent predictions, suggesting that condition-token information is processed in this layer. Importantly, the effect of patching L1 attention outputs is significantly greater for imbalanced compared to balanced, indicating that models trained under this regime rely more strongly on L1 representations to encode condition-dependent behavior.

Taken together, these results show that curricula influence both the robustness of rule-following behavior and the internal organization of rule representations. While both curricula yield similarly well-aligned surface-level behavior after fine-tuning, elicitation prompts reveal that models trained under balanced curricula are more prone to produce misaligned completions when directly probed. Mechanistically, this difference is reflected in how condition information is processed. Imbalanced training leads to more localized processing of condition information, with a strong dependence on first-layer attention (which converges faster than other layers, Fig. 25), in contrast to balanced training which distributes this computation across layers. These findings indicate that curricula shape not only model behavior, but the layer-wise organization of rule-relevant information.

4 Discussion and Conclusion

We have shown that curricula induce differences in learning dynamics, generalization, and model circuitry. These differences have important downstream consequences: For balanced curricula models are more likely to overrefuse during selective fine-tuning. Similarly, in the SLL task that includes notions of "aligned" and "misaligned" sentences, the imbalanced curriculum enabled the formation of more localized attention patterns, resulting in more effective fine-tuning for rule adherence. Notably, in the SLL task the early aligned exposure did not prevent the acquisition of misaligned behavior later in the imbalanced curriculum, but worked through the creation of representations which could be quickly re-engaged during fine-tuning. In a control curriculum in which the order of aligned to misaligned was reversed, but the imbalance was kept, we observed the loss of both L1-localization and an alignment benefit, so while we think of them as a causal chain (early aligned data producing

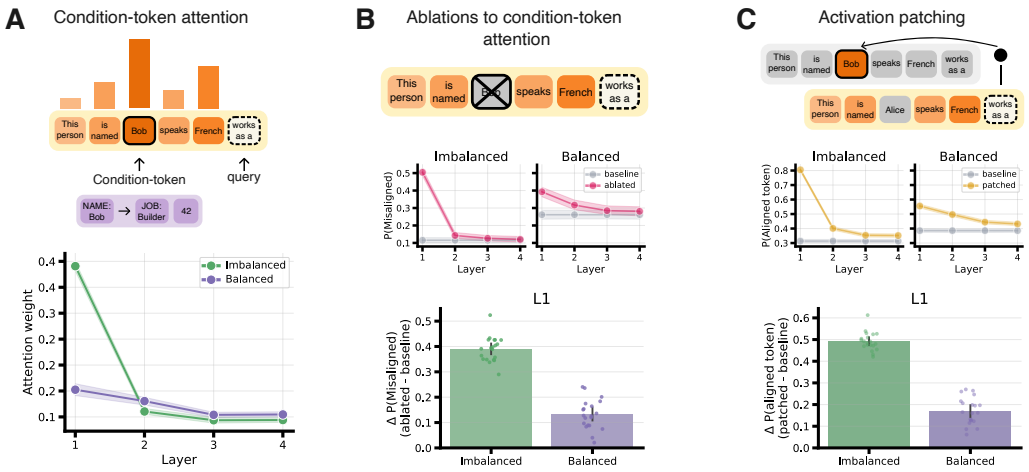


Figure 7: First layer attention is critical for rule adherence, especially under imbalanced curriculum. (**A**) Top: Schematic of attention from final token to preceding tokens. Condition-token ("Bob") specifies that a given rule applies. Bottom: Mean attention weight to condition-token across layers of transformer for imbalanced (green) and balanced (purple) models (shaded region indicates 95% CI, for all panels). (**B**) Top: Ablation of condition-token. Bottom: P(Misaligned) under ablation relative to baseline across layers (balanced vs. imbalanced). A high probability for misaligned output after ablation reveals that this attention weight is critical to rule-adherence. (**C**) Top: Activation patching from a rule-triggering source (top) sequence to a rule-neutral destination (bottom) sequence. Bottom: Probability of producing the completion that would have been required by the source sequence (P(Source target)) for patched models vs. baseline across layers. Patched values above baseline indicate that this specific attention block output drives rule-adherence.

L1 localization leading to an alignment benefit), we cannot exclude other causal mediators between curriculum and alignment. Prior work has identified that burstiness within sequences and Zipfian distributions over examples contribute to the emergence of ICL [7, 49, 55]. We similarly find that imbalanced distributions over tasks can drive ICL abilities and that learned model circuitry can differ as a function of such curricula. This identifies curricula as a potential factor in driving the formation of shared circuitry as described in the mechanistic interpretability literature [36]. LLMs mostly display overrefusal when models struggle to distinguish between subtle differences between benign and harmful content [42, 66, 62]. Our results indicate that curriculum-driven circuit entanglement may modulate this effect.

While our results show curriculum phenomena in two settings, the extent to which these phenomena are present in larger models, which have to learn a wide array of (implicit) sub-tasks, is not yet clear. There is evidence in LLMs that the implicit curricula relate to task representations [28]. Similarly, the simplified nature of our task settings potentially limits their relevance to phenomena in ethological settings. Future work should explore how selective fine-tuning differentially affects tasks with similar representations in LLMs, and how explicit pretraining curricula change learned circuitry. Our work has potential implications for pretraining: Practitioners should take the dynamics of learning into account and be aware that interference during fine-tuning may be greater for tasks trained in tandem, though confirming this at scale remains an important direction for future work.

5 Acknowledgments

We would like to thank Edmund Lau, our research sponsor, for his valuable guidance, feedback and support. We also thank Andrew Saxe, Naomi Saphra, Robert West, Julian Minder and Viktor Moskvoretskii for helpful and insightful discussions. SAB, JR, KJS, and MHW were supported by the UK AI Security Institute (AISI) through the Alignment Project Grant (AP-S2-100060). CS was supported by a Wellcome Trust Discovery Award (https://wellcome.org/grant-funding/schemes/discovery-awards) (227928/Z/23/Z).

References

[1] Z. Allen-Zhu and Y. Li. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction, July 2024. URL http://arxiv.org/abs/2309.14316. arXiv:2309.14316 [cs.CL].

[2] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, Apr. 2022. URL http://arxiv.org/abs/2204.05862. arXiv:2204.05862 [cs.CL].

[3] N. Belrose, Q. Pope, L. Quirke, A. Mallen, and X. Fern. Neural networks learn statistics of increasing complexity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 3382–3409, Vienna, Austria, July 2024. JMLR.org.

[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, June 2009. Association for Computing Machinery. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553380. URL https://dl.acm.org/doi/10.1145/1553374.1553380.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives, 2014. URL https://arxiv.org/abs/1206.5538.

[6] J. Betley, D. Tan, N. Warncke, A. Sztyber-Betley, X. Bao, M. Soto, N. Labenz, and O. Evans. Emergent Misalignment: Narrow finetuning can produce broadly misaligned LLMs. *Nature*, 649 (8097):584–589, Jan. 2026. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-025-09937-5. URL http://arxiv.org/abs/2502.17424. arXiv:2502.17424 [cs].

[7] S. Chan, A. Santoro, A. Lampinen, J. Wang, A. Singh, P. Richemond, J. McClelland, and F. Hill. Data Distributional Properties Drive Emergent In-Context Learning in Transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891, Dec. 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/77c6ccacfd9962e2307fc64680fc5ace-Abstract-Conference.html.

[8] J. Chen, X. Pan, D. Yu, K. Song, X. Wang, D. Yu, and J. Chen. Skills-in-Context: Unlocking Compositionality in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13838–13890, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.812. URL https://aclanthology.org/2024.findings-emnlp.812.

[9] M. F. Chen, N. Roberts, K. Bhatia, J. Wang, C. Zhang, F. Sala, and C. Ré. Skill-it! A Data-Driven Skills Framework for Understanding and Training Language Models, July 2023. URL http://arxiv.org/abs/2307.14430. arXiv:2307.14430 [cs.CL].

[10] A. Conmy, A. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso. Towards Automated Circuit Discovery for Mechanistic Interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, Dec. 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

[11] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits, Dec. 2021. URL https://www.transformer-circuits.pub/2021/framework/index.html.

[12] T. Flesch, J. Balaguer, R. Dekker, H. Nili, and C. Summerfield. Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, 115 (44), Oct. 2018. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1800755115. URL https://pnas.org/doi/full/10.1073/pnas.1800755115.

[13] T. Flesch, K. Juechems, T. Dumbalska, A. Saxe, and C. Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110 (7):1258–1270.e11, Apr. 2022. ISSN 08966273. doi: 10.1016/j.neuron.2022.01.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627322000058.

[14] S. Garg, D. Tsipras, P. Liang, and G. Valiant. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes, Aug. 2023. URL http://arxiv.org/abs/2208.01066. arXiv:2208.01066 [cs].

[15] M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer Feed-Forward Layers Are Key-Value Memories, Sept. 2021. URL http://arxiv.org/abs/2012.14913. arXiv:2012.14913 [cs].

[16] C. Gibson, W. Cui, and G. Reddy. Distinct mechanisms underlying in-context learning in transformers, Apr. 2026. URL http://arxiv.org/abs/2604.12151. arXiv:2604.12151 [cs.LG].

[17] M. Hanna, O. Liu, and A. Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model, Nov. 2023. URL http://arxiv.org/abs/2305.00586. arXiv:2305.00586 [cs.CL].

[18] S. Heimersheim and N. Nanda. How to use and interpret activation patching, Apr. 2024. URL http://arxiv.org/abs/2404.15255. arXiv:2404.15255 [cs.LG].

[19] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. Feb. 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

[20] D. Kalimeris, G. Kaplun, P. Nakkiran, B. Edelman, T. Yang, B. Barak, and H. Zhang. SGD on Neural Networks Learns Functions of Increasing Complexity. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/b432f34c5a997c8e7c806a895ecc5e25-Abstract.html.

[21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

[22] K. A. Krueger and P. Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, Mar. 2009. ISSN 00100277. doi: 10.1016/j.cognition.2008.11.014. URL https://linkinghub.elsevier.com/retrieve/pii/S0010027708002850.

[23] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. doi: 10.1017/S0140525X16001837.

[24] J. H. Lee, S. S. Mannelli, and A. Saxe. Why Do Animals Need Shaping? A Theory of Task Composition and Curriculum Learning, June 2024. URL http://arxiv.org/abs/2402.18361. arXiv:2402.18361 [q-bio.NC].

[25] J. H. Lee, A. K. Lampinen, A. K. Singh, and A. M. Saxe. Distinct Computations Emerge From Compositional Curricula in In-Context Learning, June 2025. URL http://arxiv.org/abs/2506.13253. arXiv:2506.13253 [cs.LG].

[26] T. Lieberum, M. Rahtz, J. Kramár, N. Nanda, G. Irving, R. Shah, and V. Mikulik. Does Circuit Analysis Interpretability Scale? Evidence from Multiple Choice Capabilities in Chinchilla, July 2023. URL http://arxiv.org/abs/2307.09458. arXiv:2307.09458 [cs.LG].

[27] Z. Lin and K. Lee. Dual Operating Modes of In-Context Learning, Aug. 2024. URL http://arxiv.org/abs/2402.18819. arXiv:2402.18819 [cs.LG].

[28] E. Liu, K. Sun, M. Li, I. Lee, L. Tjuatja, J.-t. Huang, and G. Neubig. What do Language Models Learn and When? The Implicit Curriculum Hypothesis, Apr. 2026. URL http://arxiv.org/abs/2604.08510. arXiv:2604.08510 [cs.CL].

[29] F. Locatello, S. Bauer, M. Lučić, G. Rätsch, S. Gelly, B. Schölkopf, and O. F. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019. URL http://proceedings.mlr.press/v97/locatello19a.html. Best Paper Award.

[30] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization, Jan. 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs].

[31] J. M. Mandler and L. McDonough. Concept Formation in Infancy. *Cognitive Development*, (8): 291–318, 1993.

[32] S. S. Mannelli, Y. Ivashinka, A. Saxe, and L. Saglietti. Tilting the Odds at the Lottery: the Interplay of Overparameterisation and Curricula in Neural Networks, June 2024. URL http://arxiv.org/abs/2406.01589. arXiv:2406.01589 [cond-mat, q-bio, stat].

[33] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, Nov. 2013. ISSN 1476-4687. doi: 10.1038/nature12742.

[34] F. Mathy and J. Feldman. A rule-based presentation order facilitates category learning. *Psychonomic Bulletin & Review*, 16(6):1050–1057, Dec. 2009. ISSN 1531-5320. doi: 10.3758/PBR.16.6.1050. URL https://doi.org/10.3758/PBR.16.6.1050.

[35] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and Editing Factual Associations in GPT, Jan. 2023. URL http://arxiv.org/abs/2202.05262. arXiv:2202.05262 [cs.CL].

[36] J. Merullo, C. Eickhoff, and E. Pavlick. Circuit Component Reuse Across Tasks in Transformer Language Models, May 2024. URL http://arxiv.org/abs/2310.08744. arXiv:2310.08744 [cs] version: 3.

[37] E. Michaud, Z. Liu, U. Girit, and M. Tegmark. The Quantization Model of Neural Scaling. *Advances in Neural Information Processing Systems*, 36:28699–28722, Dec. 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5b6346a05a537d4cdb2f50323452a9fe-Abstract-Conference.html.

[38] S. M. Noh, V. X. Yan, R. A. Bjork, and W. T. Maddox. Optimal sequencing during category learning: Testing a dual-learning systems perspective. *Cognition*, 155:23–29, Oct. 2016. ISSN 00100277. doi: 10.1016/j.cognition.2016.06.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0010027716301548.

[39] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. In-context Learning and Induction Heads, Sept. 2022. URL http://arxiv.org/abs/2209.11895. arXiv:2209.11895 [cs.LG].

[40] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, Mar. 2022. URL http://arxiv.org/abs/2203.02155. arXiv:2203.02155 [cs].

[41] J. Pan, T. Gao, H. Chen, and D. Chen. What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning, May 2023. URL http://arxiv.org/abs/2305.09731. arXiv:2305.09731 [cs.CL].

[42] L. Pan, Y. Tong, X. Zhang, X. Zhang, J. Zhou, and Z. Chu. Understanding and Mitigating Overrefusal in LLMs from an Unveiling Perspective of Safety Decision Boundary. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21068–21086, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1065. URL https://aclanthology.org/2025.emnlp-main.1065.

[43] R. Pastrana. Disentangling Variational Autoencoders, Nov. 2022. URL http://arxiv.org/abs/2211.07700. arXiv:2211.07700 [cs.LG].

[44] J. Pesnot Lerousseau and C. Summerfield. Shared sensitivity to data distribution during learning in humans and transformer networks. *Nature Human Behaviour*, 10(3):601–614, Mar. 2026. ISSN 2397-3374. doi: 10.1038/s41562-025-02359-3. URL https://www.nature.com/articles/s41562-025-02359-3.

[45] X. Qi, A. Panda, K. Lyu, X. Ma, S. Roy, A. Beirami, P. Mittal, and P. Henderson. Safety Alignment Should Be Made More Than Just a Few Tokens Deep, June 2024. URL https://arxiv.org/abs/2406.05946v1.

[46] P. Quirke and F. Barez. Understanding Addition in Transformers, Apr. 2024. URL http://arxiv.org/abs/2310.13121. arXiv:2310.13121 [cs.LG].

[47] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, July 2024. URL http://arxiv.org/abs/2305.18290. arXiv:2305.18290 [cs].

[48] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5301–5310. PMLR, May 2019.

[49] G. Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations*, Oct. 2023. URL https://openreview.net/forum?id=aN4Jf6Cx69.

[50] J. Rubruck, J. P. Bauer, A. M. Saxe, and C. Summerfield. Early learning of the optimal constant solution in neural networks and humans. In *Proceedings of the 8th Annual Conference on Cognitive Computational Neuroscience*, July 2025. URL https://openreview.net/forum?id=6Xyu486HRh.

[51] J. Russin, E. Pavlick, and M. J. Frank. Parallel trade-offs in human cognition and neural networks: The dynamic interplay between in-context and in-weight learning. *Proceedings of the National Academy of Sciences*, 122(35):e2510270122, 2025. doi: 10.1073/pnas.2510270122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2510270122.

[52] A. Saxe, S. Sodhani, and S. J. Lewallen. The Neural Race Reduction: Dynamics of Abstraction in Gated Networks. In *Proceedings of the 39th International Conference on Machine Learning*, pages 19287–19309. PMLR, June 2022. URL https://proceedings.mlr.press/v162/saxe22a.html.

[53] A. M. Saxe, J. L. McClelland, and S. Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, June 2019. doi: 10.1073/pnas.1820226116.

[54] S. Si, X. Wang, G. Zhai, N. Navab, and B. Plank. Think Before Refusal : Triggering Safety Reflection in LLMs to Mitigate False Refusal Behavior, Mar. 2025. URL http://arxiv.org/abs/2503.17882. arXiv:2503.17882 [cs].

[55] A. Singh, S. Chan, T. Moskovitz, E. Grant, A. Saxe, and F. Hill. The Transient Nature of Emergent In-Context Learning in Transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 27801–27819. Curran Associates, Inc., 2023. doi: 10.52202/075280-1208. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/58692a1701314e09cbd7a5f5f3871cc9-Abstract-Conference.html.

[56] A. K. Singh, T. Moskovitz, F. Hill, S. C. Y. Chan, and A. M. Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation, Apr. 2024. URL http://arxiv.org/abs/2404.07129. arXiv:2404.07129 [cs.LG].

[57] A. K. Singh, T. Moskovitz, S. Dragutinović, F. Hill, S. C. Y. Chan, and A. M. Saxe. Strategy Coopetition Explains the Emergence and Transience of In-Context Learning. June 2025. URL https://openreview.net/forum?id=esBoQFmD7v.

[58] B. F. Skinner. Reinforcement today. *American Psychologist*, 13(3):94–99, 1958. doi: 10.1037/h0049039. URL https://doi.org/10.1037/h0049039.

[59] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe. Curriculum learning: A survey. *CoRR*, abs/2101.10382, 2021. URL https://arxiv.org/abs/2101.10382.

[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. ISSN 1533-7928. URL http://jmlr.org/papers/v15/srivastava14a.html.

[61] X. Steenbrugge, S. Leroux, T. Verbelen, and B. Dhoedt. Improving Generalization for Abstract Reasoning Tasks Using Disentangled Feature Representations, Nov. 2018. URL http://arxiv.org/abs/1811.04784. arXiv:1811.04784 [cs].

[62] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023. URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs.CL].

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[64] C. Volk, C. C. Pack, and S. Bakhtiari. The curriculum effect in visual learning: the role of readout dimensionality, Sept. 2025. URL http://biorxiv.org/lookup/doi/10.1101/2025.09.19.677409.

[65] K. R. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. Sept. 2022. URL https://openreview.net/forum?id=NpsVSN6o4ul.

[66] X. Wang, C. Hu, P. Röttger, and B. Plank. Surgical, Cheap, and Flexible: Mitigating False Refusal in Language Models via Single Vector Ablation, Oct. 2024. URL https://arxiv.org/abs/2410.03415v2.

[67] A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, et al. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the babylm challenge at the 27th conference on computational natural language learning*, pages 1–34, 2023.

[68] X. Wu, E. Dyer, and B. Neyshabur. When Do Curricula Work?, Feb. 2021. URL http://arxiv.org/abs/2012.03107. arXiv:2012.03107.

[69] Z. Xue, Z. Qi, G. Liu, B. Chen, and R. Pedarsani. Deactivating Refusal Triggers: Understanding and Mitigating Overrefusal in Safety Alignment, Mar. 2026. URL http://arxiv.org/abs/2603.11388. arXiv:2603.11388 [cs].

[70] Y. Zhang, A. M. Saxe, and P. E. Latham. Saddle-to-Saddle Dynamics Explains A Simplicity Bias Across Neural Network Architectures. Oct. 2025. URL https://openreview.net/forum?id=Vit5M0G5Gb.

[71] Y. Zhang, A. Mohamed, H. Abdine, G. Shang, and M. Vazirgiannis. Beyond Random Sampling: Efficient Language Model Pretraining via Curriculum Learning, Jan. 2026. URL http://arxiv.org/abs/2506.11300. arXiv:2506.11300 [cs].

[72] N. Zucchet, J. Bornschein, S. Chan, A. Lampinen, R. Pascanu, and S. De. How do language models learn facts? Dynamics, curricula and hallucinations, Mar. 2025. URL http://arxiv.org/abs/2503.21676. arXiv:2503.21676 [cs].

Appendix Roadmap

The appendix is organized as follows:

Section	Contents
Appendix A	Extended related work: data-distributional properties in ICL, model circuits/mech. interp., refusal fine-tuning, curriculum learning, disentangled representations.
Appendix B	ICL task setup: dataset/task details, architecture & optimization, compute, pretraining curricula, refusal fine-tuning protocol.
Appendix C	Additional ICL results: learning-time differences, policy dynamics, linear probing of task identity, entanglement over fine-tuning, emergence of disentanglement over pretraining, copy-first-early curriculum, refusal accuracy, constant-imbalance curricula, infinite-data regime, batch-composition control.
Appendix D	Interpretability methods for the ICL task: mean-component ablation, direct logit attribution, activation patching (incl. full recovery table), size-matched comparison of generalizing seeds.
Appendix E	Mechanistic analyses of the refusal fine-tuned model (DLA on original vs. refusal token).
Appendix F	SLL task setup: grammar, rule system, dataset construction, curricula, architecture & optimization, compute.
Appendix G	Additional SLL results: control curricula, grammaticality checks, robustness across rule-system complexity/dataset size.
Appendix H	Mechanism analyses for SLL: attention ablations, activation patching, dynamics across training, grammaticality under intervention, attention-weight dynamics.
Appendix K	Hyperparameter sweep for the ICL task (learning rate, batch size, weight decay, width, imbalance fraction; 2- and 4-layer networks).
Appendix L	Hyperparameter sweep for the SLL task.

A Extended Related Work

Data distributional properties in ICL Previous work has shown that properties of the data distribution, such as burstiness, the frequency of occurrence of rare classes, and Zipfian data distributions are crucial for the emergence of in-context learning abilities [7, 49, 44]. Controlled function-learning studies further show that transformers can acquire in-context learning algorithms when trained on suitable distributions over tasks or functions [14]. However, how the emergence and structure of model circuits differ as a function of learning curricula has not been explored in much detail.

Model circuits in mechanistic interpretability Mechanistic interpretability seeks to identify the circuits responsible for specific model behaviors [17, 10, 26, 65]. For example, the mechanisms underlying induction circuits [39] in simple attention-based models are now understood in considerable detail [49, 56, 57, 11]. At the same time, circuits initially thought to be task-specific can be gainfully reused across distinct tasks [36]. However, when and why models reuse circuitry across sub-tasks, and when they instead develop separated, sub-task specific circuitry [37, 46], remains poorly understood. We propose that curricula, implicit or explicit, are one driver of whether sub-tasks come to share mechanisms or are implemented by more separated computations.

Refusal fine-tuning in LLMs The alignment of LLMs for helpfulness and harmlessness is typically implemented via a pipeline that includes supervised fine-tuning and preference optimization such

as reinforcement learning from human feedback and direct preference optimization [40, 2, 47]. In supervised safety fine-tuning, models are directly trained on adversarial prompts with corresponding safe demonstrations [62, 69, 45]. However, overrefusal where LLMs erroneously refuse benign queries can in some cases undermine the practical utility of these techniques [62, 69]. The reasons for such overrefusal have been explored from several angles. For example, previous work has suggested these behaviors are likely when prompts are close to safety decision boundaries where models struggle to distinguish between harmful and harmless user requests [42] and have been found to be induced by specific linguistic cues [69]. Mitigation techniques include the ablation of "overrefusal directions" [66] or through the targeted prompting with specific safety reflections [54]. While previous work has focused on representational factors driving overrefusal, our work seeks to explain how learning dynamics and curricula can give rise to entangled task representations that can give rise to unintended overgeneralization behaviors.

Curriculum learning Human and animal learning is greatly aided by the ordered presentation of information [22, 58] and there have been considerable efforts to translate this benefit into techniques for machine learning [4]. In spite of this, curriculum learning has proven non-trivial to successfully implement for neural networks [67, 68] and has not become an established part of the training pipeline of frontier LLMs [71]. Classic curriculum learning proceeds by gradually increasing the difficulty of presented training examples, which means it requires: (i) a difficulty criterion for ranking examples by difficulty, and (ii) a scheduler, for determining when an example of which difficulty level should be shown [59]. While superficially straightforward, this hides deep questions such as how to define difficulty, whether it is objective or student dependent, how the scheduler should manage sample diversity, and how it should space training samples to minimize forgetting. Curricula are therefore a tantalizing tool, but there are no generally reliable guidelines for their implementation yet.

Disentangled representations Disentangled representations are those that decompose factors of variation, i.e. they disentangle the individual elements of the environment or task, rather than representing multiple components in a combined, entangled manner [5]. Prior work has proposed that disentangled representations support learning success, generalization [61], robustness, and interpretability [23, 19], and reduce the impact of catastrophic forgetting [21]. Locatello et al. [29] demonstrated that disentangled representations cannot be reliably learned in fully unsupervised settings without inductive priors or supervision signals, suggesting that training structure may play a role in encouraging disentanglement. Humans preferentially form disentangled representations when trained in a blocked manner, i.e. focusing on a single task for an extended period [12, 13]. Finding curricula which promote disentangled representations for neural networks remains an area of active research.

B Experimental setup for the ICL task

B.1 Task and dataset

Each example string is a combination $k = 5$ tokens (drawn i.i.d. with replacement from a 50-symbol vocabulary), followed by a colon, a single-symbol label, and a separator. A training sequence concatenates $n = 4$ labeled context examples followed by an unlabeled query (feature + colon); cross-entropy is applied at the query colon. Total sequence length is 38 tokens. The vocabulary has 53 tokens (50 symbols + colon + separator + a reserved refusal token). Each sequence is assigned a single task by a per-sequence Bernoulli draw and all $n + 1$ examples follow that task; we resample features within a sequence until all $n + 1$ are distinct. The training set is fixed at $S = 20{,}000$ sequences revisited each epoch. The evaluation set is 500 sequences sampled at $50/50$ with a different seed.

Task	Context examples	Query	Target
Copy-first (CF)	`kqzbn:k | fhdsi:f | owrxc:o | vutpl:v`	`jaegm:`	`j`
	`tymzq:t | leohr:l | dwnub:d | pivfg:p`	`ckxas:`	`c`
	`rxhqj:r | mceyk:m | bdozi:b | sfglz:s`	`nvtwo:`	`n`
Copy-last (CL)	`hzbyk:k | rcweo:o | pqxnu:u | vgmds:s`	`fjlta:`	`a`
	`oebmq:q | ucpiv:v | nrdsa:a | ywhpz:z`	`gxlkt:`	`t`
	`tphwq:q | jvbom:m | keurd:d | scfan:n`	`yzigl:`	`l`

Table 2: Illustration of example sequences for each task. Each sequence concatenates n=4 in-context examples followed by an unlabeled query. An example is a feature of k=5 symbols, a colon, a label symbol, and a separator (|). Under *copy-first* the label is the first symbol of the query; under *copy-last* the label is the last symbol. The model is trained to predict the correct label at the colon following the query (rightmost column). Symbols are drawn from a 50-symbol token vocabulary

B.2 Architecture and optimization

We use a 2-layer Transformer with $d_{\text{model}} = 128$, with 4 heads per layer of dimension $d_{\text{model}}/4 = 32$, GELU FFN of hidden dimension $4\,d_{\text{model}} = 512$, learned positional embeddings and dropout 0.1 [60]. Pretraining: AdamW [30], β_1=0.9, β_2=0.999, lr 3×10^{-4}, weight decay 0.01, gradient-norm clipping at 1.0, batch size 64. All models were trained using PyTorch.

B.3 Compute

We used 3 NVIDIA GeForce RTX 3090 GPUs for this study. For the main experiments in Section 2 training a single seed and a single curriculum for experiment 1 took around 20 minutes of GPU time, leading to a total compute amount of 100 hours (100 seeds for 3 curricula).

B.4 Pretraining curricula

Three conditions: (i) imbalanced, copy-last early (10:90); (ii) balanced (fixed at 50:50 task mixture); (iii) imbalanced, copy-first early (90:10, Appendix C.6). For imbalanced conditions, the copy-first fraction is held at the starting value for the first 10 epochs of pretraining, linearly faded to 50:50 between 10 and 90 epochs, and held at 50% thereafter for 30 epochs. We train 100 seeds per condition for the main experiments; the checkpoint analyses of Appendix C.5 use an earlier set of 30-seed runs, as noted there. Analyses comparing curricula are restricted to seeds achieving $\geq 90\%$ held-out accuracy on both tasks.

B.5 Refusal fine-tuning

Starting from the final pretrained checkpoint, we fine-tune on a 90% refused-task / 10% kept-task mixture: refused-task labels to the final query are replaced with the refusal token, kept-task targets to the final query are unchanged. AdamW with lr 5×10^{-5}, 3 epochs over a fresh 20,000-sequence training set (distinct seed from pretraining); other hyperparameters match pretraining. We evaluate

the models on a 50:50 evaluation. We report refusal rate (fraction of last-position predictions equal to the refusal token) and kept task accuracy computed per task.

C Additional results for the controlled in-context learning task

C.1 Learning time difference between curricula

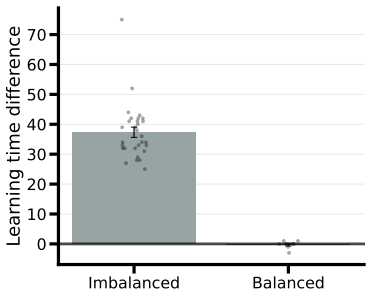


Figure 8: Difference in learning time in epochs between the two sub-tasks, time of copy-first minus time of copy-last. For each seed, we computed the first epoch at which copy-first and copy-last each exceeded 90% test accuracy, retaining only seeds that reached threshold on both tasks. Bars show the mean signed difference, with standard-error error bars and individual seeds overlaid.

C.2 Development of model policy

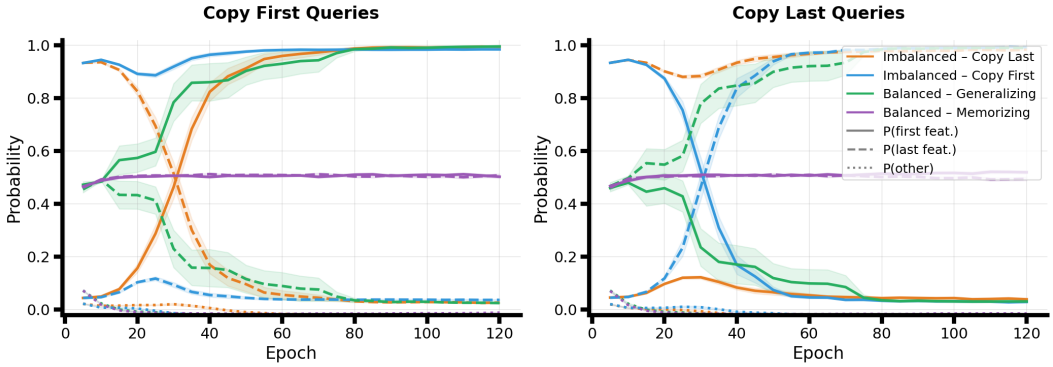


Figure 9: Probability distribution of the model over different groups of outputs (copy-first consistent, copy-last consistent, other). Lines show the mean, shaded regions indicate ± s.e. (*left*) Probability distribution outputted by different models over the course of training on held-out copy first queries. Models trained on the (light blue) imbalanced copy-first condition learn to quickly allocate almost all of their probability mass on (solid line) the first token of the sequence, and very little on (wide dashed line) the last token and even less on (thin dashed lines) all other tokens. Models trained on the (orange) imbalanced copy-last condition start out allocating most of their probability mass to the last token, even on this condition of copy first queries, but then learn to differentiate smoothly. Models trained in the balanced condition evenly spread their probability mass between both the first and last tokens. They are divided into two groups, (green) generalizing models that slowly learn to accurately distinguish between first and last tokens and (purple) memorizing models that remain randomly guessing between the first and last solution over the whole of training. (*right*) Same as left, but instead for copy last queries. Shaded region represents error of mean.

C.3 Linear probing for task identity

To quantify how readily the two subtasks are linearly represented in the residual stream after pre-training, we trained a logistic-regression probe on activations taken at the query position after each

sublayer: the embedding output, after L1 attention, after L1 FFN, after L2 attention, and after L2 FFN. For each trained model, we constructed two single-task probe datasets, one from copy-first examples and one from copy-last examples, and evaluated probe accuracy using cross-validation (500 new task samples per task). Probe accuracy near 50% indicates little linear separability, while higher values indicate that the two sub-tasks are linearly represented. We find that task identity is only strongly linearly represented in the later layers of models that are trained in the imbalanced curriculum and support our hypothesis around circuit entanglement in the balanced curriculum.

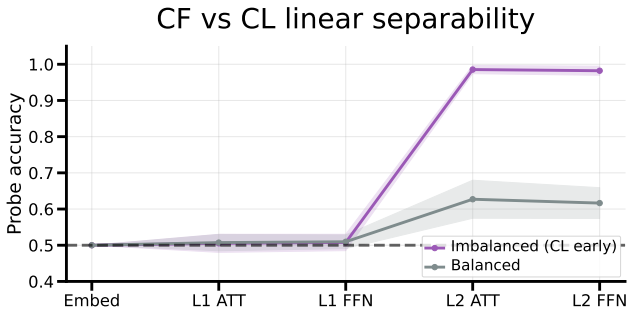


Figure 10: Task identity is linearly represented only for imbalanced models. Linear probe accuracy for discriminating copy-first versus copy-last representations across sub-layers. Curves show mean probe accuracy across seeds for each training regime (shaded bands: $\pm$ s.d.). We find that only for models trained in the imbalanced curriculum task identity is strongly represented linearly after pretraining.

C.4 Curriculum-dependent task entanglement can be measured directly

Homing in on the previous result, we were interested in the similarity of residual stream activity for the two tasks across the refusal fine-tuning period. To this end, we computed the cosine similarity between these two task representations after the different curricula and while fine-tuning against different tasks, see Fig. 11. Interestingly, this reveals that the curricula create vastly different starting configurations: Under the balanced curriculum, task representations are highly correlated at the outset, and only gradually decorrelate over the course of fine-tuning. Task representations under the imbalanced curriculum are nearly orthogonal (disentangled) at the start of refusal training, and then only slightly change in their cosine similarity during fine-tuning. This strongly suggests that the highly entangled representations under the balanced curriculum contribute to the observed overrefusal. Additionally, it suggests that monitoring the angle between related tasks during fine-tuning might serve as an early warning signal more generally, to detect whether fine-tuning has unwanted side effects on other capabilities.

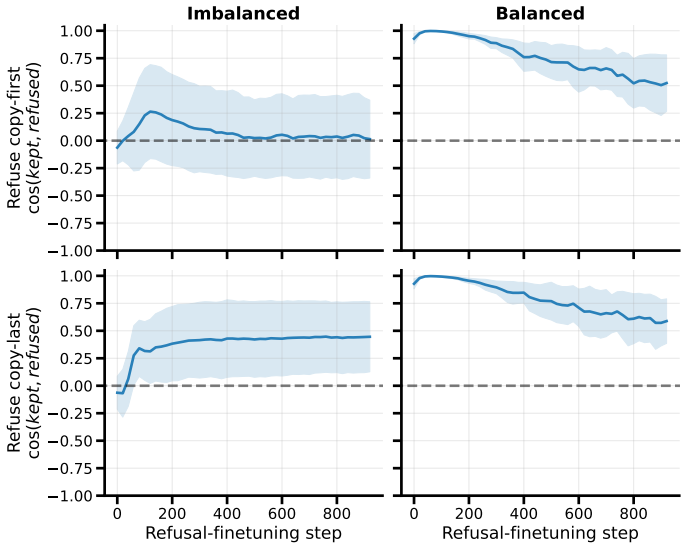


Figure 11: Residual stream task directions for the two tasks are closer to orthogonal under imbalanced curricula. We show the cosine similarity between residual stream activity at the colon position at the final L2 FFN of the refused and the kept task during fine-tuning on paired prompts that only flip the task label. The columns split this comparison by the different curricula, rows by whether we refusal-train copy-first or copy-last. We focus on residual stream activity after the final FFN block (lines represent the mean across seeds, shaded regions indicate $\pm$s.d.). Under a balanced curriculum, the distance between task representations starts at close to one, maximal correlation, and then decreases during fine-tuning. For imbalanced curricula on the other hand, the distances start orthogonal, around zero. This provides some explanations as to why refusal fine-tuning more strongly affects the kept task after a balanced curriculum.

C.5 Disentanglement emerges gradually over pretraining

The analyses in Section 2.3 characterize the circuitry of the *converged* model, which leaves open whether the task-specific circuits found under the imbalanced curriculum are present from the outset of training or form progressively. We therefore repeat two of our measures every 5 pretraining epochs, evaluating each checkpoint on a fresh held-out set of 500 sequences sampled at a 50:50 task ratio. We include every seed that generalizes by the end of pretraining ($30/30$ imbalanced and $8/30$ balanced seeds; these analyses use the 30-seed runs instead of the 100-seed in the main).

Mean-ablation across checkpoints We mean-ablate L2 ATT at each checkpoint, exactly as in Appendix D.1, and record held-out accuracy per task (Fig. 12). At the end of pretraining the converged dissociation of Fig. 3B is recovered and L2 ATT removes copy-first (0.02) but spares copy-last (0.99) under the imbalanced curriculum, while both tasks fall to chance (0.52 and 0.43) under the balanced curriculum.

Task cosine across checkpoints We also track the cosine similarity between the residual stream task representations at the final L2 FFN, the same quantity reported over fine-tuning in Fig. 11, taken at the final query colon over paired prompts whose in-context labels are flipped to the other task's rule (Fig. 13). Under the imbalanced curriculum the two task directions start entangled (cosine ≈ 1.0) and separate to near-orthogonal (-0.07) over pretraining, with an onset around epochs 30–35; under the balanced curriculum the cosine plateaus around 0.9 for the whole of pretraining, including for the seeds that generalize. The endpoint of this trajectory is the configuration that refusal fine-tuning starts from, i.e. step 0 of Fig. 11.

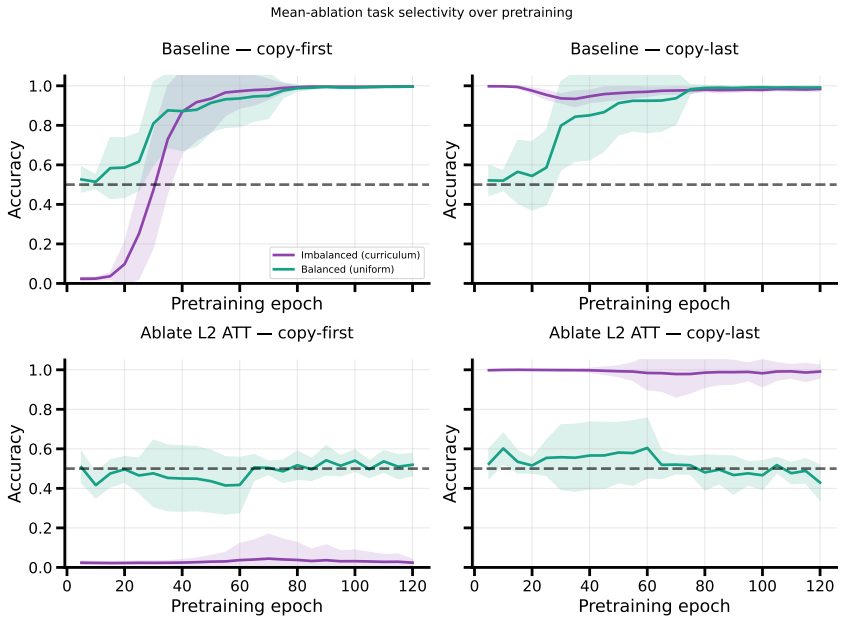


Figure 12: The task-selective circuit forms gradually and only under the imbalanced curriculum. Held-out accuracy per task at every pretraining checkpoint for imbalanced (*purple*, $n = 30$) and balanced (*green*, $n = 8$) seeds that generalize by the end of pretraining (lines show the mean across seeds, shaded regions indicate $\pm$ s.d., dashed line marks chance). *Top*: unablated baseline accuracy on copy-first (*left*) and copy-last (*right*), showing when each task is acquired. *Bottom*: the same accuracies after mean-ablating L2 ATT. Under the imbalanced curriculum, ablating L2 ATT leaves copy-last at ceiling while destroying copy-first; under the balanced curriculum the same ablation drives both tasks to chance throughout training.

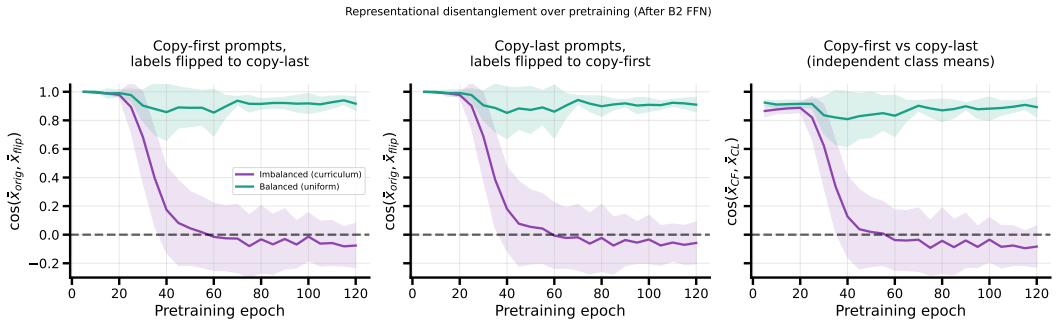


Figure 13: Task representations orthogonalize over pretraining only under the imbalanced curriculum. Cosine similarity between residual stream task representations at the final L2 FFN, at the query colon, across pretraining checkpoints (same seeds, colors and conventions as Fig. 12; dashed line marks orthogonality). *Left* and *center*: the paired measure of Fig. 11, comparing a set of single-task prompts to the same prompts with their in-context labels flipped to the other task's rule, for copy-first and copy-last originals respectively. *Right*: the cosine between independently drawn copy-first and copy-last class means. On all three measures the imbalanced curriculum moves from entangled (≈ 1.0) to near-orthogonal (≈ -0.07) with an onset around epochs 30–35, whereas the balanced curriculum plateaus around 0.9 throughout pretraining.

C.6 Inverse imbalance: Copy-first early

Acquisition dynamics. For completeness, we also consider an imbalanced learning setting in which the copy-first task is introduced early in training. We find similar results for learning dynamics such that the imbalanced training helps models to consistently learn a generalizing solution Fig. 14.

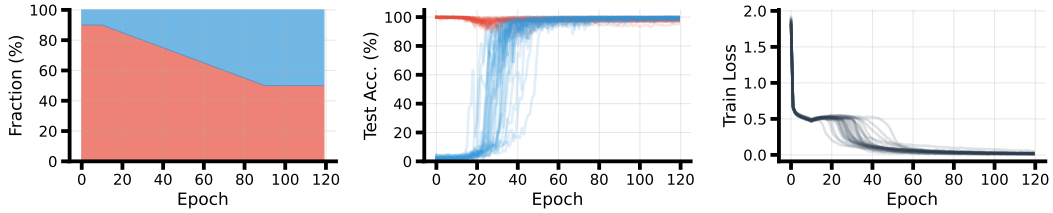


Figure 14: Training dynamics for the imbalanced copy-first-early curriculum. Copy first is trained early, the task mixture is gradually annealed to 50:50. Left: task fraction over training. Center: held-out test accuracy on both tasks across seeds. Right: training loss. Under this curriculum, all $30/30$ seeds converge to generalizing solutions.

Refusal results We further find that models in this curriculum show similar effects during the refusal phase as seen in Fig. 15. For models in this type of imbalanced curriculum, the refusal rate on the unrefused (kept) task similarly remains low throughout fine-tuning.

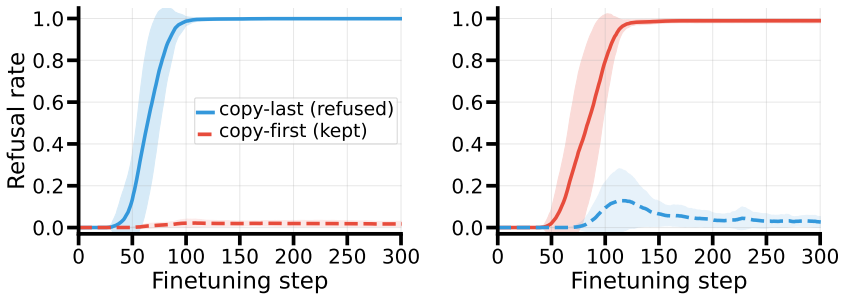


Figure 15: Refusal training introduces only minimal interference in the inverse imbalanced curriculum. Same as Fig. 2 but for the copy-first early imbalanced curriculum in Fig. 14, lines represent the mean with shaded regions indicating standard deviation over seeds. We find that models trained under this type of imbalanced curriculum similarly show less interference.

C.7 Accuracy during refusal fine-tuning

Beyond refusal rates we also examine accuracy on the kept (non-refused) task throughout fine-tuning. We find that accuracy on the held-out tasks does not decrease significantly during refusal fine-tuning for the imbalanced curriculum, and is degraded but then recovers for the balanced curriculum (Fig. 16). These results exactly mirror observations in Fig. 2 but show that models trained under imbalanced curricula do not only not over-refuse but also retain performance on the original task, while performance in balanced models temporarily degrades due to overrefusals.

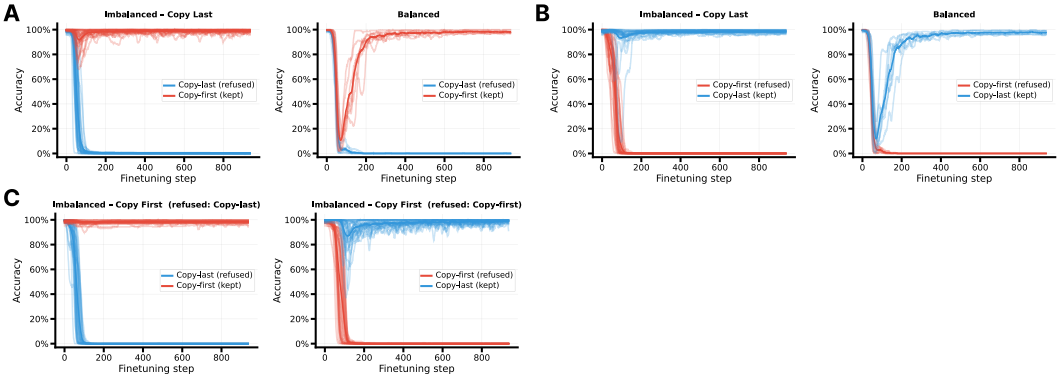


Figure 16: Accuracy on the kept task is maintained in imbalanced models over the course of refusal fine-tuning. (**A**) Accuracy while refusal fine-tuning the copy-last task for both (*left*) imbalanced curricula and (*right*) balanced curricula from the main text. (**B**) Same as (A), but refusal fine-tuning on the copy-first task. (**C**) Results for the imbalanced curriculum in which the copy-first task is introduced early in training while refusal fine-tuning on (*left*) copy-first and (*right*) copy-last.

C.8 Constant imbalanced training data distributions

Acquisition dynamics We further consider imbalanced training data distributions that do not contain explicit linear fading of the task mixture but manipulate the order of task acquisition via two fixed constant task mixtures of 15:85 and 85:15 (copy first:copy last). We train 5 seeds per condition with all other hyperparameters as in Section 2 and Appendix B.1. We find that this manipulation similarly encourages the learning of generalizing solutions as seen in Fig. 17 such that all $5/5$ seeds learn to generalize (>90% accuracy). These results imply that task imbalance beyond fading can also encourage generalization. However, we note that seeds in this regime converge slower than in Fig. 1. These results should be seen as a form of implicit curriculum where the data distribution indirectly encourages the sequential learning of both sub-tasks. It is also noteworthy that these results rule out that the dynamic resampling of tasks seen in the imbalanced curricula is the driver of improved generalization behavior. We further find that models in this regime also show reduced overrefusals during fine-tuning (see Fig. 18).

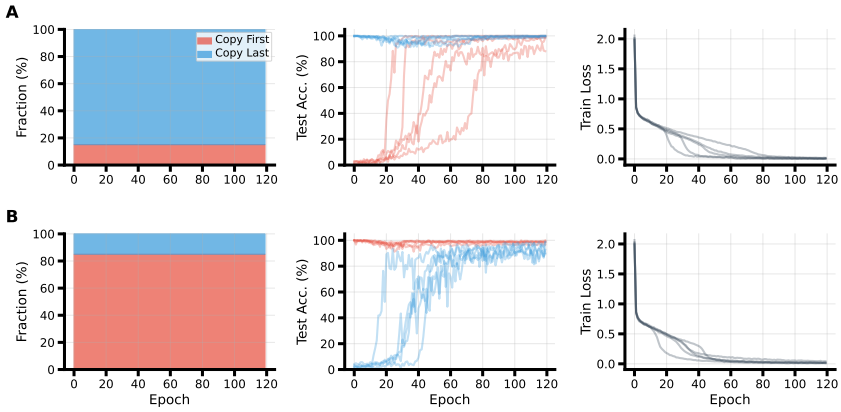


Figure 17: Training dynamics for constant imbalanced datasets. Imbalanced training tasks without linear fading also allow models to learn generalizing solutions albeit at a slightly slower rate (compare to Fig. 1) Left: task fraction over training. Center: held-out test accuracy on both tasks across seeds. Right: training loss. Under this curriculum, all $5/5$ seeds converge to generalizing solutions.

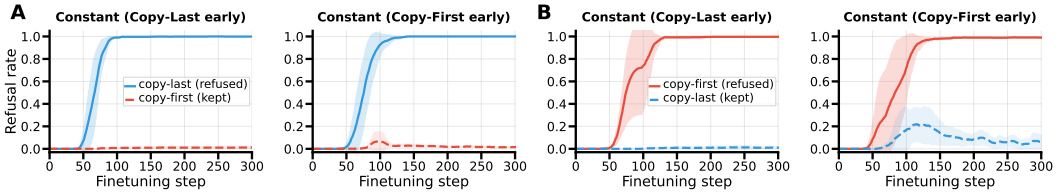


Figure 18: Refusal training introduces only minimal interference in the constant imbalanced regime. Same as Fig. 2 but for constant imbalanced task mixtures shown in Fig. 17, lines represent the mean with shaded regions indicating standard deviation over seeds. We find that models trained under this type of imbalanced curriculum similarly show reduced interference when selectively unlearning one of the two sub-tasks.

C.9 Training with infinite data

The results in the main paper used a fixed training set of $S = 20{,}000$ sequences that the model revisits every epoch. However, given the importance of data distributional properties for the emergence of ICL [7, 49, 55] we also evaluate models trained in a regime of *infinite* data where we sample new sequences every epoch. Concretely, at the start of each epoch t the training set is re-sampled, yielding a fresh draw of $S = 20{,}000$ sequences. All other Hyperparameters are identical to the fixed-data runs. This increases the effective diversity of training samples, ensuring the model cannot overfit to a fixed pool of samples and instead must learn the underlying task structure directly.

Figure 19B shows that surprisingly, models trained under the balanced curriculum do not learn the generalizing solution such that $0/20$ seeds converge to a generalizing solution within 120 epochs, while for the imbalanced regime (Fig. 19A) $19/20$ seeds acquire a generalizing ICL solution. These results underscore that imbalanced curricula also reliably produce generalizing solutions in the infinite data regime. However, it is perhaps surprising that in our multi task setting greater diversity reduced the propensity of models to successfully acquire ICL solutions in the balanced curriculum. Previous results have demonstrated that greater data diversity typically facilitates the acquisition of ICL solutions [7]. In contrast, the result in this section shows that for task recognition modes of ICL [41] perhaps the partial acquisition of memorizing, in-weights solutions facilitates in-context learning in a more cooperative manner [57].

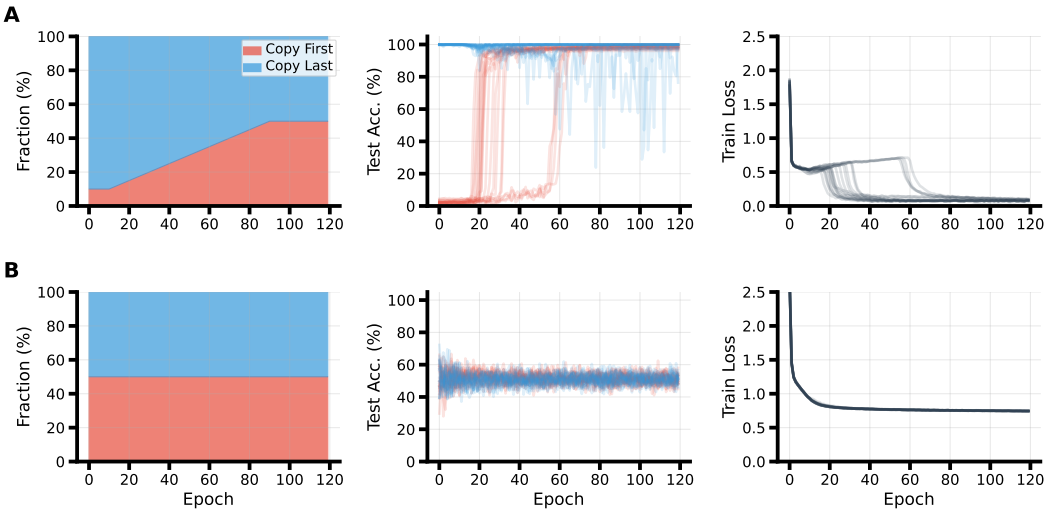


Figure 19: Training curricula in the infinite data regime. Same as Fig. 1 but for infinite data.

C.10 Robustness to batch composition

We have attributed the circuit differences of Section 2.3 to the task proportions set by the curriculum. One possible confound is the purity of the training batches. Under the balanced curriculum the gradient of every batch mixes both tasks, whereas the imbalanced curriculum spends its early epochs on batches that are nearly pure. To separate the proportion schedule from reduced task-mixing in the gradient signal, we repeated pretraining with fully homogeneous (single-task) batches under both curricula. Each batch is drawn entirely from one task, and the curriculum sets the fraction of copy-first *batches* per epoch so the overall task proportions and the number of optimizer steps are unchanged. We train 30 seeds under the imbalanced and 100 under the balanced curriculum to account for their differing generalization rates, and as elsewhere restrict the analyses to generalizing seeds ($30/30$ and $10/100$ respectively).

The imbalanced curriculum reproduces the full dissociation of Fig. 3 under homogeneous batching ($n = 30$): mean-ablating L2 ATT removes copy-first ($2.2 \pm 0.9\%$) but spares copy-last ($99.7 \pm 1.0\%$), and copy-last is carried by L2 FFN (DLA 8.56 ± 2.67). Reduced task-mixing is therefore not necessary for the task-specific circuit to form. Neither is it sufficient: under the homogeneously batched balanced curriculum ($n = 10$) the copy-last-specific L2 FFN pathway is absent (DLA

0.65 ± 0.61 versus the imbalanced curriculum, Welch $t = 14.8$, $p < 10^{-4}$) and ablating L2 ATT yields only a partial and highly variable separation (copy-first $36.9 \pm 17.7\%$, copy-last $62.8 \pm 18.9\%$). The behavioral consequence reproduces as well: peak overrefusal on the kept task stays low under the imbalanced curriculum (2.0–3.2%) but is substantial under the balanced curriculum (36.5–41.2%), mirroring Fig. 2. We note that the balanced curriculum generalizes less often under homogeneous than under mixed batching ($10/100$ versus $22/100$), which we attribute to the task being constant within a homogeneous batch, so that a constant output-position bias suffices to reduce that batch's loss and the in-context labels become redundant for fitting it.

D Interpretability analyses: the controlled in-context learning task

Both analyses are performed on each pretrained checkpoint and evaluated on the held-out 500-sequence 50:50 test set described in Appendix B.1. As elsewhere in the paper, analyses are restricted to seeds that reach $\geq 90\%$ held-out accuracy on both tasks at the end of pretraining; results are reported per task (copy-first vs copy-last) and per curriculum.

D.1 Mean-component ablation

For each candidate sublayer (L1 ATT, L1 FFN, L2 ATT, L2 FFN), we first compute the position-wise dataset-mean activation of that sublayer's output over the evaluation set. We then re-run the model with that sublayer's output replaced by its mean (a mean ablation that preserves the global activation statistics but removes input-conditional information, [65]), and measure per-task accuracy on the same held-out set. The unablated baseline is included for comparison. Reported figures focus on attention block ablations.

D.2 Direct logit attribution

Direct logit attribution (DLA) decomposes output logits into the direct contributions of components that write to the residual stream [11, 65]. We apply DLA at the query colon position. Let h denote the final pre-LayerNorm residual stream at this position, decomposed as $h = \sum_i c_i$, where each c_i is the embedding contribution or the output of an attention/FFN sublayer. Since the final unembedding is applied after the final LayerNorm, we include the learned LayerNorm scale γ and the per-token residual-stream standard deviation $\sigma(h)$. Treating $\sigma(h)$ as fixed at its forward-pass value, the direct contribution of component c_i to the correct-token logit y is

$$\mathrm{DLA}_i(y) = \left[(c_i/\sigma(h))(W_U\gamma)^\top\right]_y .$$

This fixed-scale approximation makes the final LayerNorm linear in the residual-stream components, so component contributions are additive and approximately reconstruct the correct-token logit. We report per-component contributions on the held-out set, separately for copy-first and copy-last queries.

D.3 Activation patching

Activation patching is a technique that carefully replaces specific internal activation of transformers during a forward pass from a previous run [18, 35]. In this paper we use this technique to causally reveal which model components are implicated in the solution of the learned copy-first and copy last sub-tasks. Here we specifically use *denoising* activation patching to localize the pathways through which each sub-task is implemented.

Corruption. For each "clean" sequence drawn from one task, we construct a "corrupt" counterpart by recomputing the label of every in-context example under the *other* task's rule; all features (including the query feature) are left unchanged. Under the clean input the model follows the in-context rule and predicts the correct label. Under the corrupt input it follows the flipped rule and predicts the other task's answer, dropping accuracy on the original target to near chance.

What is patched. A denoising patch runs the forward pass on the corrupt (label flipped) input but overwrites one or more activations with the corresponding values from the clean forward pass; every other activation is recomputed from the corrupted input. For L2 attention we patch the sublayer's Q-, K-, and V-projections individually, pairwise, and triple combinations (QK, KV, QV and QKV restoration). For L2 FFN we patch the residual read (the residual stream immediately before the FFN sublayer). While the main paper only reports patching results from L2 Q, K, V, and FFN as they provide the decisive contrast, we conduct and report the patching of combinations for completeness.

Recovery metric. For each patch we compute recovery as

$$\frac{\mathrm{Acc}_{\mathrm{patched}} - \mathrm{Acc}_{\mathrm{corrupt}}}{\mathrm{Acc}_{\mathrm{clean}} - \mathrm{Acc}_{\mathrm{corrupt}}},$$

separately for copy-first and copy-last queries. A recovery of 0 corresponds to no improvement over the corrupt baseline; 1 corresponds to full restoration of clean-run accuracy.

Imbalanced

Patch	CF Acc	CL Acc	CF Rec.	CL Rec.
Clean	99.6 ± 0.5	98.4 ± 1.1	—	—
Corrupt	3.8 ± 1.6	2.8 ± 1.2	(base)	(base)
L2 Q	3.8 ± 1.6	2.8 ± 1.2	0.0 ± 0.0	0.0 ± 0.0
L2 K	81.6 ± 20.0	39.8 ± 32.7	81.2 ± 20.7	39.0 ± 34.4
L2 V	22.5 ± 21.7	21.2 ± 20.9	19.7 ± 22.8	19.3 ± 21.6
L2 QK	81.6 ± 20.0	39.8 ± 32.7	81.2 ± 20.7	39.0 ± 34.4
L2 KV	99.6 ± 0.5	98.4 ± 1.1	100.0 ± 0.0	100.0 ± 0.0
L2 QV	22.5 ± 21.7	21.2 ± 20.9	19.7 ± 22.8	19.3 ± 21.6
L2 QKV	99.6 ± 0.5	98.4 ± 1.1	100.0 ± 0.0	100.0 ± 0.0
L2 FFN	56.2 ± 35.5	88.1 ± 22.7	54.9 ± 37.1	89.3 ± 23.8

Balanced

Patch	CF Acc	CL Acc	CF Rec.	CL Rec.
Clean	99.6 ± 0.4	99.2 ± 0.7	—	—
Corrupt	2.7 ± 0.7	2.5 ± 1.2	(base)	(base)
L2 Q	2.7 ± 0.7	2.5 ± 1.2	0.0 ± 0.0	0.0 ± 0.0
L2 K	99.4 ± 0.4	98.4 ± 0.9	99.8 ± 0.3	99.3 ± 0.6
L2 V	3.4 ± 1.0	3.0 ± 1.3	0.7 ± 0.5	0.5 ± 0.6
L2 QK	99.4 ± 0.4	98.4 ± 0.9	99.8 ± 0.3	99.3 ± 0.6
L2 KV	99.6 ± 0.4	99.2 ± 0.7	100.0 ± 0.0	100.0 ± 0.0
L2 QV	3.4 ± 1.0	3.0 ± 1.3	0.7 ± 0.5	0.5 ± 0.6
L2 QKV	99.6 ± 0.4	99.2 ± 0.7	100.0 ± 0.0	100.0 ± 0.0
L2 FFN	2.9 ± 0.7	2.6 ± 1.2	0.2 ± 0.2	0.1 ± 0.2

Table 3: Denoising activation patching across all evaluated channels. Accuracies and recovery fractions are reported in percent as mean $\pm$ standard deviation across seeds.

Full table. Table 3 reports every patch we evaluated under both training conditions. "Clean" and "Corrupt" rows give raw accuracy. Rows give the accuracy and recovery fraction obtained when a given channel is patched; Q,K,V-combinations (e.g. "Restore QK") patch multiple projection inputs jointly.

Why does restoring L2 KV recover copy-last under Imbalanced training, when copy-last has been argued to bypass Block 2 attention? For imbalanced training, mean ablation (Section 2.3) shows that *removing* L2 attention leaves copy-last largely intact, so the sublayer is not *necessary* for copy-last. Denoising patching with restored KV (Table 3) shows that supplying clean K and V to L2 attention *recovers* copy-last to clean-run accuracy, so the sublayer is *sufficient* to perform copy-last. These two statements are not in conflict: L2 attention has the capacity to solve copy-last but does not need to, because L1 attention already writes the copy-last answer into the residual stream that L2 FFN reads. The two-pathway claim is therefore about *primary routing*, not about hard capability boundaries: copy-first is primarily routed through K (L2 ATT) and copy-last through the FFN residual read, but both sublayers retain residual capacity to solve either task. The Balanced condition lacks this redundancy: restoring the FFN residual read alone yields essentially zero recovery on either task, so all routing flows through K.

D.4 Size-matched comparison of generalizing seeds

Because the curricula generalize at different rates, restricting the analyses above to generalizing seeds leaves the two arms with unequal sample sizes ($100/100$ imbalanced versus $22/100$ balanced). To rule out a sample size confound we subsampled the imbalanced pool down to the balanced sample size and recomputed every quantity in Fig. 3B and C, drawing $20{,}000$ subsamples of $n = 22$ seeds and reporting the subsample mean with a 95% percentile interval.

The size-matched estimates are indistinguishable from the full-pool estimates (Table 4): every size-matched mean agrees with the $n = 100$ mean to two decimal places, and no balanced value falls inside the corresponding interval. The same holds for direct logit attribution. Under the balanced

curriculum L2 ATT contributes almost identically to both tasks (12.53 ± 1.24 for copy-first versus 12.57 ± 1.22 for copy-last) and L2 FFN to neither (0.53 ± 0.21 and 0.53 ± 0.22), whereas the imbalanced curriculum splits them: L2 ATT carries copy-first (14.54 ± 3.09) but not copy-last (0.35 ± 0.17), while copy-last is carried by L2 FFN (16.30 ± 2.01). Size-matching leaves both unchanged (L2 ATT copy-first 14.54 $[13.07, 15.57]$; L2 FFN copy-last 16.30 $[15.35, 17.01]$). The circuit-level dissociation is therefore not an artifact of the smaller balanced sample.

Ablation	Task	Balanced ($n = 22$)	Imbalanced ($n = 100$)	Imbalanced size-matched ($n = 22$)
Baseline	Copy-first	99.6 ± 0.5	99.6 ± 0.6	99.6 $[99.4, 99.8]$
Baseline	Copy-last	99.5 ± 0.5	98.8 ± 1.1	98.8 $[98.3, 99.2]$
L1 ATT	Copy-first	13.1 ± 17.9	2.2 ± 0.9	2.2 $[1.8, 2.5]$
L1 ATT	Copy-last	11.9 ± 14.4	2.2 ± 1.0	2.2 $[1.8, 2.6]$
L2 ATT	Copy-first	51.3 ± 6.6	2.7 ± 6.1	2.7 $[1.7, 5.2]$
L2 ATT	Copy-last	50.3 ± 6.4	98.4 ± 7.0	98.4 $[94.7, 100.0]$

Table 4: Size-matching the imbalanced pool leaves the ablation results unchanged. Held-out accuracy (%) per task after mean-ablating each attention block, for the generalizing seeds of each curriculum. The first two columns report mean $\pm$ s.d. over all available seeds; the third subsamples the imbalanced pool to the balanced sample size (20,000 draws of $n = 22$), reporting the subsample mean with a 95% percentile interval in square brackets.

E Mechanistic analyses on refusal fine-tuned model

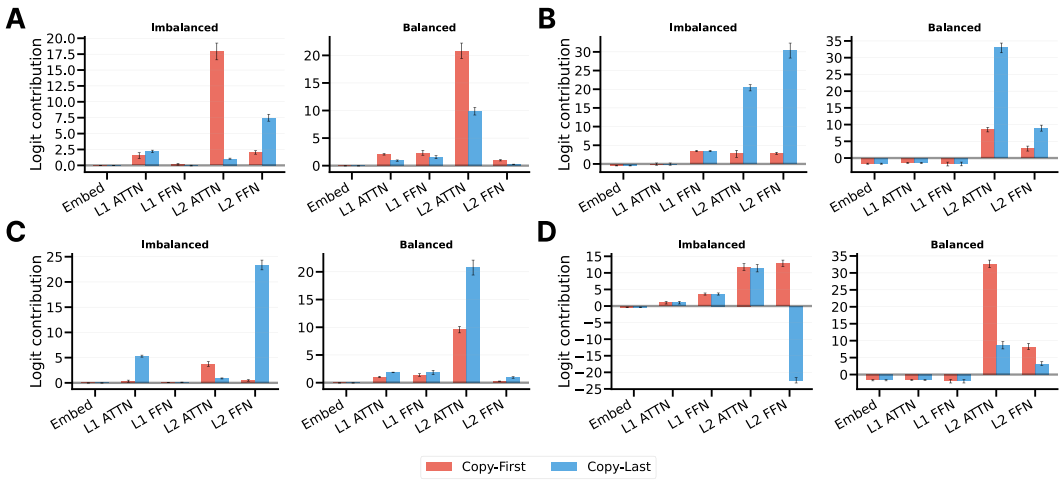


Figure 20: DLA analysis on refusal fine-tuned models reveals that refusal is primarily implemented in L2 components. (**A**) Logit contributions for the correct token under the original task of different components for models fine-tuned to refuse the copy-last task. Models were pretrained under curricula that were (left) balanced and (right) imbalanced. (**B**) Same as (A), but on the refusal token. (**C**) Same as (A), but for models fine-tuned to refuse the copy-first task. (**D**) Same as (C), but for the refusal token. Error bars depict the standard error of the mean.

F Experimental setup for the SLL task

F.1 Synthetic language grammar

We construct a synthetic language of profile-style sentences from a fixed grammar over features. Each profile sentence begins with "This person" followed by a number of feature descriptions, for example: "This person is named Bob, lives in Paris, speaks French, has a pet lizard, and drinks tea".

For a given feature, a value is selected from a finite pool (e.g. NAME $\in$ {Alice, Bob, Charlie, Diana}). Each feature value is paired with a preamble (e.g. "is named Bob", "is called Bob"). A feature value can be combined with any corresponding feature preamble, and the features can appear in any arbitrary order in the sentence.

In this experiment, the grammar consists of 9 features, and each feature was associated with a finite pool of 4 values and 2 preambles. The sentences have a fixed length of 5 features, such that the general form is:

$$\text{This person} < \text{Preamble}_1 > < \text{Value}_1 > ... < \text{Preamble}_5 > < \text{Value}_5 >$$

We used a discrete vocabulary in which each preamble (e.g. "is named") and feature value (e.g. "Bob") corresponds to a unique token. We do not replicate the full system here, since any set of unique tokens can be used to represent the feature values.

F.2 Rule system

On top of the grammar, we construct a system of rules to operationalize the concept of alignment for our task. Rules take the form of specifying a condition (e.g. "NAME=Bob") and a target (e.g. "JOB=Builder"). If a sentence meets the condition of a rule, it also needs to meet the target, otherwise the sentence is considered misaligned. If a sentence does not specify the targeted feature then the sentence counts as aligned by default. A sentence is considered aligned only if it satisfies all applicable rules. If two rules clash by specifying different target values for the same feature, the higher priority rule takes precedence.

Rule	Condition	Target	Priority
R_1	NAME = Alice	JOB = Builder	10
R_2	SPORT = Tennis	LOCATION = Sydney	5
R_3	PET = Rat	MUSIC = Rock	7
R_4	MUSIC = Pop	LOCATION = Chicago	15
R_5	PET = Crane	JOB = Doctor	18

Table 5: SLL task rule system. We employed 5 rules, two pairs of which had the potential to clash (R_1 and R_5 on JOB, R_2 and R_4 on LOCATION.)

The grammar and rule system are independent, a sentence can therefore be either: ungrammatical, if it violates the grammar, grammatical but misaligned, if it follows the grammar but not the rule system, or grammatical and aligned, if it follows both. Additionally, we draw a distinction between explicitly aligned sentences (those that trigger at least one rule and satisfy all triggered rules) and neutral sentences (those that are aligned by default as they do not trigger any rules).

F.3 Dataset construction and splits

We randomly sample 400 neutral, 400 explicitly aligned (rule condition and target met), and 400 misaligned sentences for training.

In order to probe whether the model has learned each rule and can generalize to unseen prompts, we construct a test set of single rule elicitation prompts, each triggering solely one rule. We sample 50 elicitation prompts for each rule (250 total). To ensure that none of these prompts appear in the training set, we exclude any sentences containing the elicitation prompts from the training set.

F.4 Curricula

During pretraining, models are trained on a mixture of aligned and misaligned sentences. At each epoch t, a sampling distribution over these datasets is defined by the weight assigned to misaligned examples, $w_{\text{misaligned}}(t)$. The remaining probability mass is split evenly between explicitly aligned and neutral data.

$$w_{\text{aligned}}(t) = w_{\text{neutral}}(t) = \frac{1 - w_{\text{misaligned}}(t)}{2} \tag{1}$$

We consider two curriculum conditions:

Balanced The proportion of misaligned data is held constant throughout pretraining:

$$w_{\text{misaligned}}(t) = \rho, \tag{2}$$

where ρ is the corruption level (here $\rho = 0.4$).

Imbalanced The proportion of misaligned data increases over training. We use a piecewise schedule with an initial plateau followed by a linear ramp:

$$w_{\text{misaligned}}(t) = \begin{cases} w_{\text{low}}, & t < T_{\text{plateau}} \\ w_{\text{low}} + (w_{\text{high}} - w_{\text{low}}) \cdot \tau(t), & t \geq T_{\text{plateau}}, \end{cases} \tag{3}$$

where $\tau(t)$ linearly interpolates from 0 to 1 over the remaining epochs.

To ensure curricula are matched for the total exposure to misaligned data, we choose w_{high} such that the average proportion of misaligned data matches the target corruption level:

$$\frac{1}{T} \sum_{t=1}^{T} w_{\text{misaligned}}(t) = \rho. \tag{4}$$

We train for 800 epochs on the pretraining distribution, followed by 200 fine-tuning epochs on a distribution with no misaligned examples, a 50:50 mixture of explicitly aligned and neutral data.

F.5 Architecture and optimization

We use a pre-norm 4-layer Transformer with $d_{\text{model}} = 128$, 4 heads per layer (head dim 32), a GELU FFN with hidden dimension $4\, d_{\text{model}} = 512$, rotary positional embeddings, and dropout 0.1. Optimization uses Adam with learning rate 5×10^{-4}, batch size 100, and gradient-norm clipping at 1.0. We train 20 random seeds and save checkpoints every 40 epochs during pretraining. All models were trained in PyTorch.

F.6 Compute

We used NVIDIA GeForce RTX 3090 GPUs for this study. For the main SLL experiments, training a single seed and a single curriculum took just under three minutes of GPU time, leading to a total compute time of two GPU hours (20 seeds times 2 curricula in the main text).

G Additional results for the SLL task

G.1 Grammaticality of free generations and elicitations

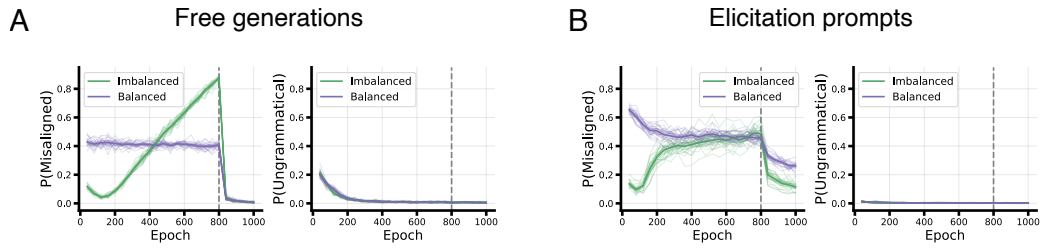


Figure 21: Misalignment and grammaticality of free generations and elicitations. P(Misaligned) (left) and P(Ungrammatical) (right) for free generations (**A**) and elicitation prompts (**B**) across training under the imbalanced (green) and balanced (purple) curricula (thin: individual seeds; bold: mean; dashed line: start of fine-tuning).

G.2 Elicitation prompt alignment with rule system complexity and data set size

To establish the robustness of our finding that fine-tuning was improved after imbalanced training, we varied the complexity of the rule system and data set size, extending beyond the fixed training set used for the main text. We left all attributes of the rule system and training process untouched, except:

1. We increased the number of values (Alice, Bob, ...) for each feature (NAME, JOB, ...) from 4 to 15, to allow for more rule variety.
2. We vary the number of rules (from 5 to 40), and randomly sample that number of rules for each run independently.
3. We vary the number of sentences making up the data set, while keeping the number of sentences presented during each "epoch" the same (1200, which is the total data set size after combining 400 neutral, aligned, and misaligned sentences in the original data set).

When sampling rules, we ensure that each condition (e.g. NAME=Alice) occurs in at most one rule, but do not control the rule target, other than ensuring that it targets a different feature than the condition (as a result, multiple rules may demand the same specific target feature).

For our results we simultaneously vary the number of rules (rule complexity) and the number of sentences in the three groups (neutral, aligned, misaligned), the data set size, to study the efficacy of fine-tuning along this spectrum, see Fig. 22.

G.3 Robustness to batch composition

We have attributed the differences in attention localization to the differing training group proportions between the balanced and imbalanced curriculum. However, one possible confound is the purity of the training batches: in the balanced curriculum, the gradient signals for "aligned" and "misaligned" samples are fully mixed, whereas the imbalanced curriculum will usually train on batches which provide a clearer gradient signal. To control for this difference, we repeated our training procedure using fully homogeneous batches in both curricula. This means that rather than composing a batch out of $X\%$ of aligned samples and $Y\%$ of misaligned samples (with X and Y determined by the curriculum), a batch is fully one or the other, and the probability for which type of sentence makes up the batch being determined by the curriculum. In this way each batch gives a clear gradient, but the overall type proportions are upheld.

The effects of the curriculum upon later fine-tuning are still very strong: P(misaligned) after fine-tuning is 0.095 ± 0.05 (mean ± SD), under imbalanced and 0.216 ± 0.02 under balanced, compare Fig. 6F. Our findings on the layerwise localization of task-relevant circuitry reproduce as well. These results indicate that the proportion schedule drives the effect.

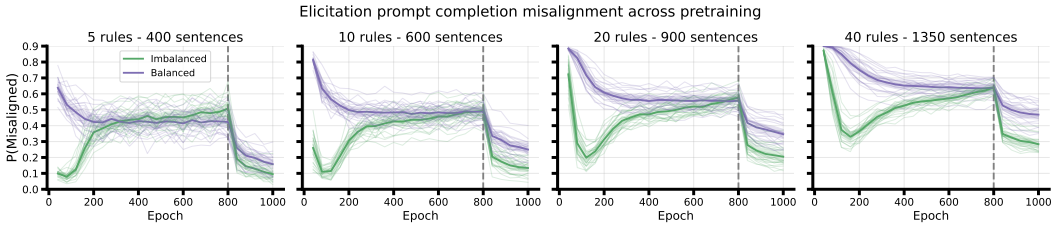


Figure 22: Alignment on elicitation prompts is more responsive to fine-tuning after imbalanced training with more rules and sentences. We replicate the elicitation dynamics plot of Fig. 6E, varying the number of rules and number of sentences of the data set. That is, we show P(Misaligned) for the elicitation prompt completions during training and fine-tuning, for balanced and imbalanced curricula. The leftmost figure closely replicates the original setting (though rather than the same five rules it samples them for each of the 20 runs). For the other plots we linearly increase the number of rules within the data set, as well as the number of sentences comprising the training set. Performance reaches comparable levels at the end of training across curricula, but, for all settings, fine-tuning is clearly more effective after imbalanced training.

H Mechanism analyses: SLL task

H.1 Attention ablations

To test the effect of condition tokens upon correct (aligned) token prediction, we ablated attention in a highly targeted manner. We zeroed the attention weight from the final query-token to the condition-token and renormalized the other. That is, for a given layer L we set:

$$\tilde{\alpha}^{(L)}_{t_{\text{final}},j} = \begin{cases} 0 & \text{if } j = c \\ \frac{\alpha^{(L)}_{t_{\text{final}},j}}{\sum_{k \neq c} \alpha^{(L)}_{t_{\text{final}},k}} & \text{otherwise} \end{cases} \tag{5}$$

where α and $\tilde{\alpha}$ denote the attention weights before and after ablation respectively, with its superscript indicating layer and subscript indicating from where to where attention is pointing, and c denotes the condition-token position. This intervention selectively removes condition-token information flow at a given layer while minimally perturbing other attention weights, allowing us to test whether attention from the final token to the condition-token is necessary for rule-following behavior.

H.2 Activation patching

For each single-rule elicitation prompt, we construct a destination prompt with an alternative condition value such that the rule no longer applies. For example, given the sequence: "This person is named *Bob* and works as a ...", which triggers the rule: "IF NAME=Bob $\rightarrow$ JOB=Builder", we construct the destination sequence as: "This person is named *Alice* and works as a ...". We expect P(Source Target) (i.e. P(Builder)) to be at baseline for the destination as the rule no longer applies. For a given layer L, we cache the output of the attention block (not attention weights, but the final output to the residual stream) for a source prompt and replace the destination's final token attention output with the corresponding source activation:

$$\tilde{a}^{(L)}_{destination}[t_{final}] = a^{(L)}_{source}[t_{final}] \tag{6}$$

where $\tilde{a}$ denotes the attention output after patching (since we exchange the full attention output we suppress token indices from the subscript, but use it to denote source and destination prompt). As the source and destination differ only in the condition-token, this intervention tests whether condition-token information encoded in the final token attention output at layer L is sufficient to drive prediction of the source's aligned token.

H.3 Interpretability analyses across training

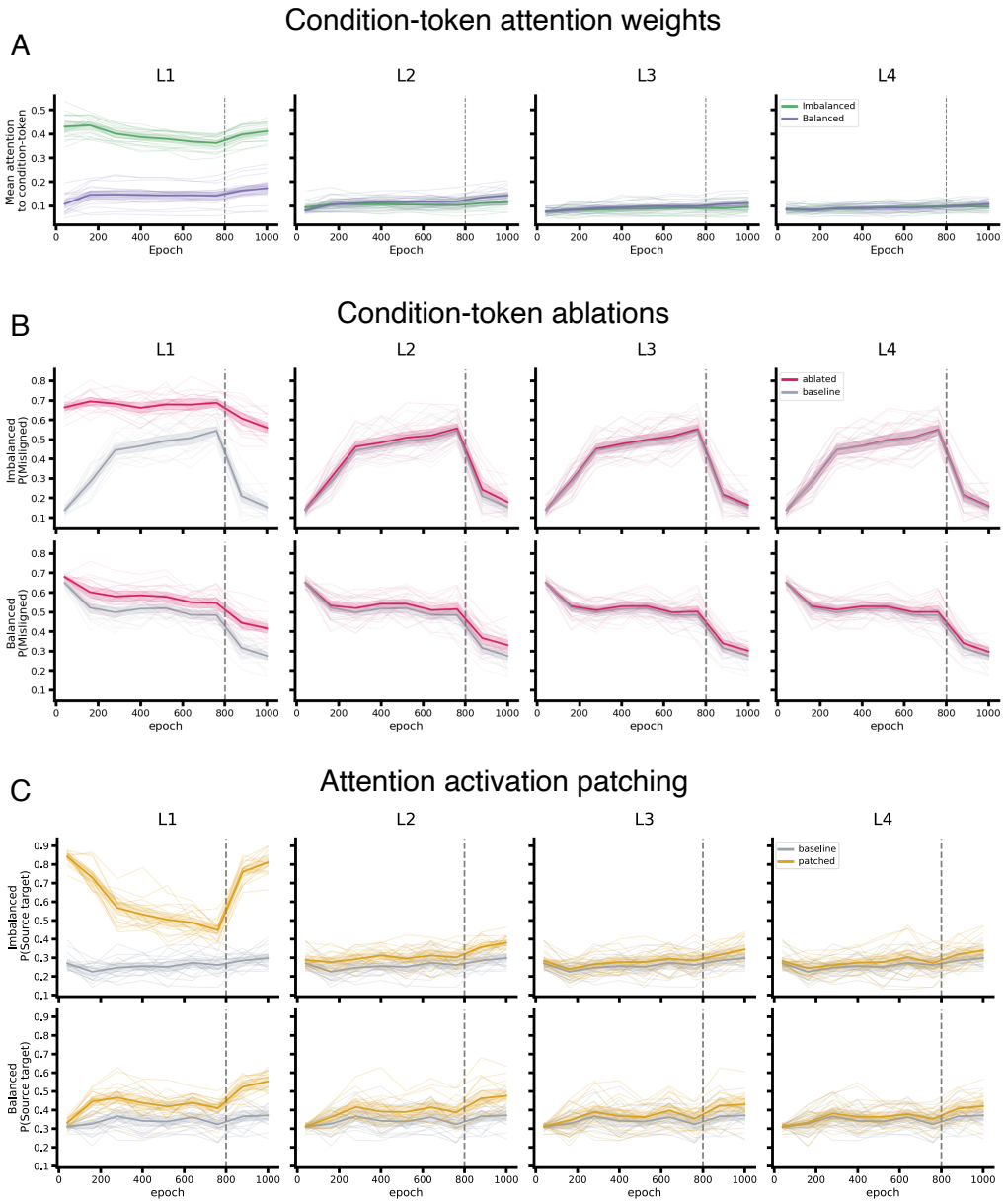


Figure 23: Development of condition token attention during pretraining. (A) Condition-token attention weights. Attention from the final token to condition-token position across layers as a function of training epoch (thin: individual seeds; bold: mean; shaded region: 95% CI; dashed line: start of fine-tuning, for all panels). Under the imbalanced curriculum (green) attention to condition-token in L1 emerges earlier and remains high across training compared to balanced (purple). **(B) Condition-token ablations** Effect of ablating attention from final token to condition-token position at each layer across epochs. L1 ablations (pink) increase P(Misaligned) relative to baseline (gray), this effect emerges earlier and is greater for models trained under imbalanced (top) relative to balanced (bottom) curricula. **(C) Attention activation patching.** Patching the attention block output at a given layer and final-token position is replaced with that from a rule-triggering source sequence. Patching L1 representations increases P(Source target), showing that these early-layer representations are sufficient to steer predictions. The effect emerges earlier and more strongly under imbalanced (top) curricula relative to balanced (bottom).

H.4 Misalignment and grammaticality under ablation and patching

Above we explored how the model produces rule-aligned output in response to a rule-triggering elicitation prompt, by ablating and patching attention patterns. This revealed that the attention from the query-token to the condition-token in the first layer is critical for a rule conforming output (ablations) and the output of that same attention block is also sufficient to strongly drive a particular rule adhering output in a different sentence (activation patching). This importance of the first layer was especially strong under the imbalanced curriculum, highlighting its distinct role. An important caveat for this analysis is to assess whether the model was perturbed more generally by the manipulations, which could affect our interpretation of them. We therefore also evaluated the probability of the model to produce altogether ungrammatical completions in response to our experiments, Fig. 24. As shown by the figure, however, no condition led the model to exhibit an increase in the prevalence of ungrammatical completions. This establishes that the observed effects of the attention pattern are mainly about rule-aligned behavior.

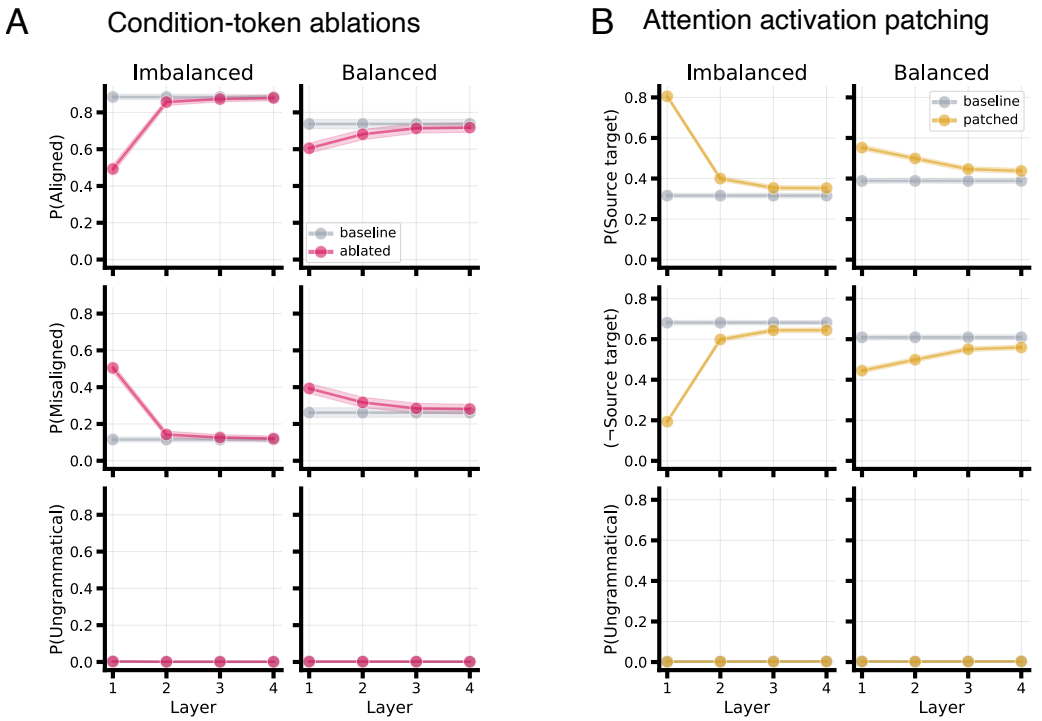


Figure 24: Attention ablation and activation patching on condition-tokens do not affect grammaticality. Extended version of Fig. 7. (**A**) Effects of zeroing attention from the query-token to the rule condition-token on mean model performance, separated by model layers (shaded region indicates 95% CI, for all panels). Top and center row: As discussed, disturbing the attention of early layers impacts the ability of the model to produce a rule-aligned output. Bottom row: However, the ablations do not at all affect whether the model is able to produce a well-formed output completion, i.e. the probability for producing an ungrammatical sentence stays close to 0 for all manipulations. (**B**) Impact of patching the attention block output from a rule-triggering sentence into a rule-neutral sentence. Top and center row: As previously shown, patching the attention block output of early layers suffices to drive rule-adherence in an otherwise neutral sentence. Bottom row: Again, however, the ability of the model to produce a grammatical output is not affected by this manipulation.

I SLL attention weight dynamics

Interpretability analyses indicate that rule-following behavior depends strongly on L1 attention. Under imbalanced curriculum, both attention to the condition-token and the effect of causal interventions on L1 are more pronounced. To investigate whether this corresponds to layer-wise differences in learning dynamics, we measure the cosine similarity between QKV attention weights at each epoch and their final values, Fig. 25. We find that L1 weights converge earlier in training relative to later layers, under both imbalanced and balanced curricula. This suggests that L1 representations are already relatively stable at later stages of pretraining. Consequently, imbalanced curricula, which prioritize aligned data early, may lead to more robust encoding of rule-condition dependencies, while balanced curricula distribute this signal more across layers.

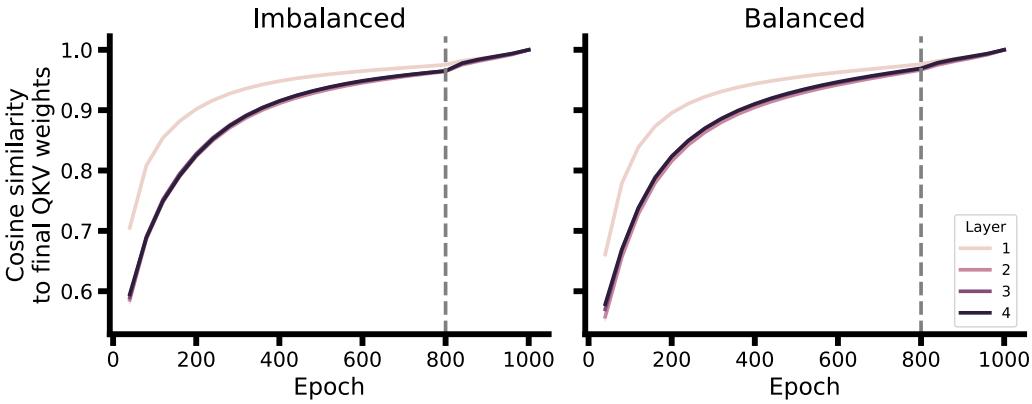


Figure 25: First layer QKV attention weights stabilize earlier than later layers. Mean cosine similarity between QKV attention weights at each epoch and their final values for each layer, under imbalanced (left) and balanced (right) curricula, shaded regions indicating 95% CI are included but not visible. Layer 1 converges faster compared to later layers. Gray dashed line marks onset of fine-tuning

J Additional control SLL curricula

J.1 Additional curricula

Previously we compared two curricula (balanced and imbalanced) matched in total exposure to misaligned data. We now compare these to three additional controls, (Fig. 26A):

- **Imbalanced_reverse**: The proportion of misaligned data decreases from 80% to 5% across pretraining, i.e. the reverse of the imbalanced curriculum.
- **Balanced_low**: A constant low proportion of misaligned data (5%)
- **Balanced_high**: A constant high proportion of misaligned data (80%)

J.2 Alignment behavior

We first examine P(Misaligned) of freely generated sentences. This closely tracks the training distribution across curricula, and fine-tuning results effectively suppresses misaligned generations in all cases with negligible differences between them, Fig. 26B.

We next analyze P(Misaligned) for elicitation prompts during pretraining. Balanced_high and balanced_low converge to high and low plateaus respectively, while imbalanced_reverse decreases gradually, mirroring the reversal for the imbalanced curriculum. These trends qualitatively reflect the training distribution.

Fine-tuning leads to a reduction in P(Misaligned) for all curricula, however, the magnitude of improvement varies. Imbalanced and balanced_low achieve similarly low levels of misalignment, suggesting limited exposure to misalignment early in training is as effective as maintaining low exposure throughout training. Furthermore, imbalanced_reverse closely matches balanced_high after fine-tuning, indicating that the impact of early exposure to high levels of misaligned data is not mitigated by cleaner data later in training. Interestingly, imbalanced_reverse performs better than imbalanced at the end of pretraining, yet fine-tuning is less effective at reducing misalignment. Together these results highlight that early training has lasting effects on alignment that do not get overwritten by later training or fine-tuning.

J.3 Mechanistic analyses

We next repeated the mechanistic analyses across all five curricula. Patterns in mean condition-token attention weight (Fig. 26A), ablations (Fig. 26B), and activation patching (Fig. 26C) are similar for curricula with the same early exposure to misaligned data. Specifically, balanced_low resembles imbalanced, and balanced_high resembles balanced and imbalanced_reverse.

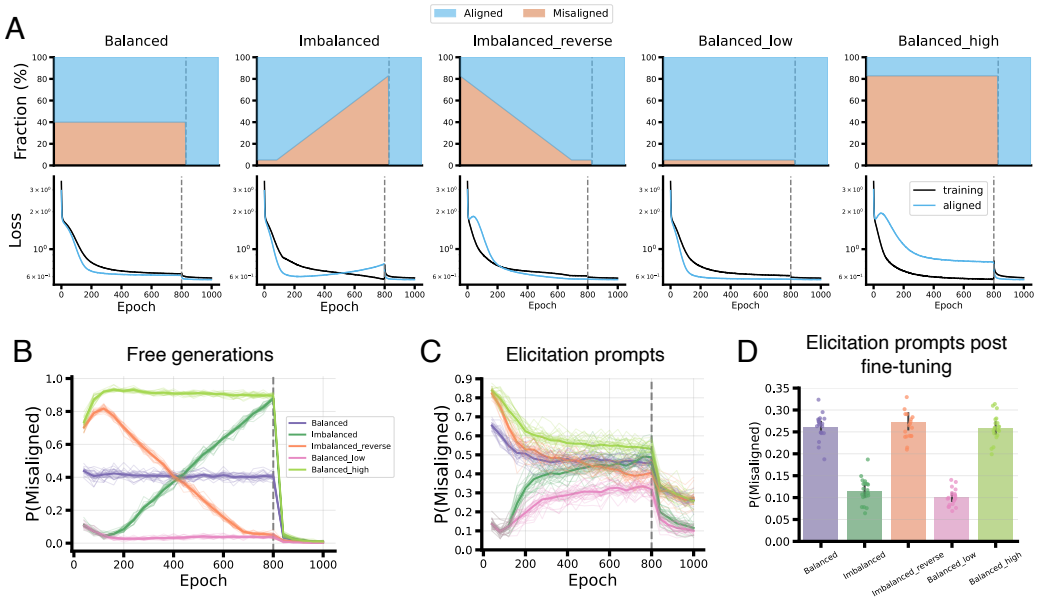


Figure 26: Additional curricula for SLL task. (**A**) Top row: To properly control our curriculum manipulation, we train models under additional regimes, beyond the balanced and imbalanced one presented in the paper. Critically, the imbalanced reverse curriculum starts with a large proportion of misaligned data, which decreases over time, flipping the order of the imbalanced curriculum. Bottom: Loss on the full training data (dark blue) and the aligned subset (light blue), shaded regions indicating 95% CI are included but not visible. (**B**) Probability of misaligned sentences during free generations under different curricula. This probability tracks the epoch-to-epoch prevalence of misaligned data under the curriculum relatively closely (thin: individual seeds; bold: mean; dashed line: start of fine-tuning). (**C**) Elicitation prompts, however, reveal different degrees of robustness of rule-adherence (same plotting convention as B). (**D**) Mean probability of a misaligned completion for elicitation prompts under different curricula. After fine-tuning, the imbalanced curriculum performs on par with a curriculum which encountered a low level of misaligned data throughout the entirety of training (balanced low). On the other hand, the reverse imbalanced curriculum performs as poorly as balanced high, which experienced a constantly high proportion of misaligned data (error bars indicate 95% CI over seeds; individual seeds: dots).

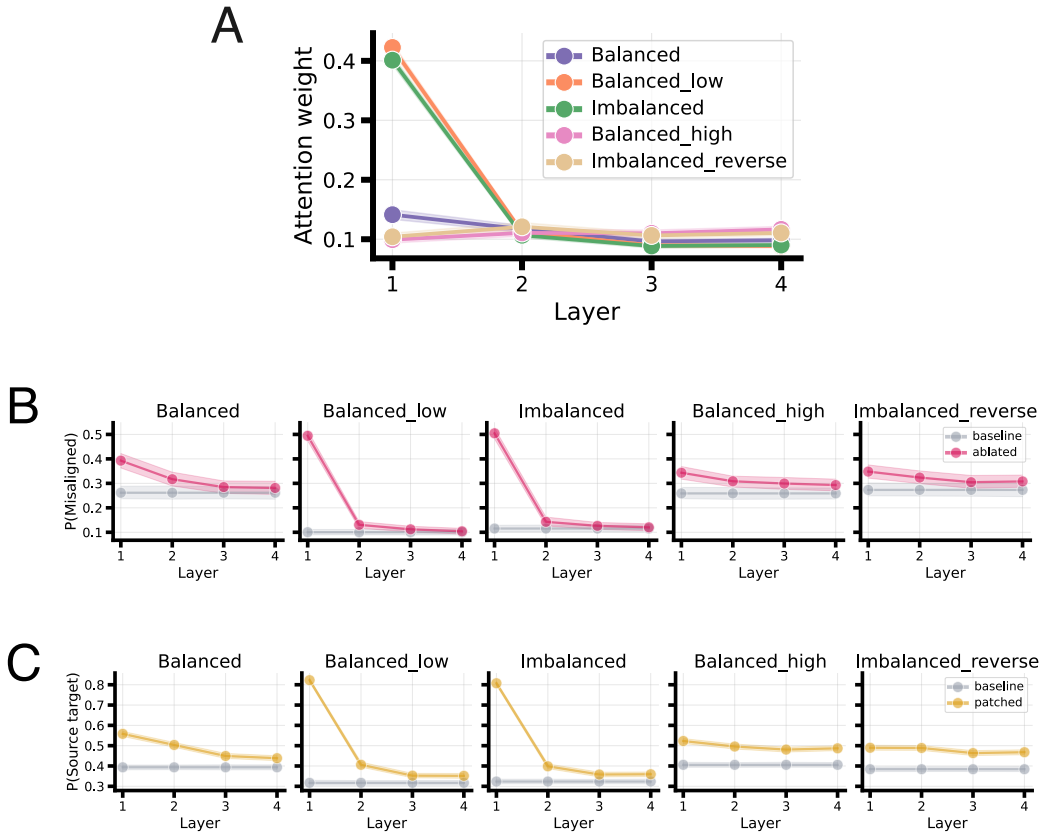


Figure 27: Mechanistic analyses across five curricula. (**A**) Mean attention weight to the condition-token across layers for transformers trained under the five curricula. (**B**) P(Misaligned) under ablation (magenta) relative to baseline (gray) across layers. (**C**) Probability of producing the completion that would have been required by the source sequence (P(Source target)) for patched models (yellow) vs. baseline (gray) across layers.

K Hyperparameter sweep for the controlled in-context learning task

To validate the robustness of our findings, we present results for different hyperparameters. We vary learning rate, batch size, weight decay, and width of the network, as well as the imbalance fraction of the curriculum. Rather than running a full grid, we vary one of these hyperparameters at a time, keeping the others fixed at the values they have for the experiments presented in Section 2. This generates different slices through the grid. In addition to doing this for the two-layer networks presented in the main text, we also repeat the whole set of experiments for networks with four layers. We train 10 seeds for each hyperparameter setting. We omit results on imbalanced curricula that start with copy-first early as results are broadly symmetric.

For each hyperparameter setting, we evaluate the fraction of runs reaching 90% test accuracy to establish generalization (Figures 28 and 32 for two and four layers, respectively); Learning speed (Figures 29 and 33); difference in accuracy after ablating different components to measure component specialization (Figures 30 and 34); and maximum refusal on the untrained task during refusal fine-tuning to evaluate overrefusal (Figures 31 and 35).

Collectively, although there are individual hyperparameter settings where the networks also generalize consistently in the balanced conditions (see in particular Fig. 28A), these settings are substantially more rare compared to the ones where generalization occurs for the imbalanced condition, for both two and four layer networks. Therefore, imbalanced curricula broadly help generalization. Interestingly, we also see emergence of occasional component specialization for certain hyperparameters in the balanced condition. These specialized seeds also show lower overrefusal during fine-tuning. Indeed, correlating specialization and overrefusal rates across all conditions and hyperparameters reveals a negative correlation for both two and four layers (see Fig. 36). We note that, as the formation of network circuitry like an induction head requires two layers [39], the task requires at least two layers to solve; indeed, in experiments we conducted with a single layer, the network was unable to solve the task.

Additionally, the hyperparameter sweep allows for a matched-convergence control: At lr=0.001 with 30 seeds each, both curricula reach 30/30 generalization, but peak overrefusal is still higher for the balanced condition at 17.3 ± 2.5 compared with 5.2 ± 0.5 for the imbalanced one ($t = 4.684$, $p < 0.001$).

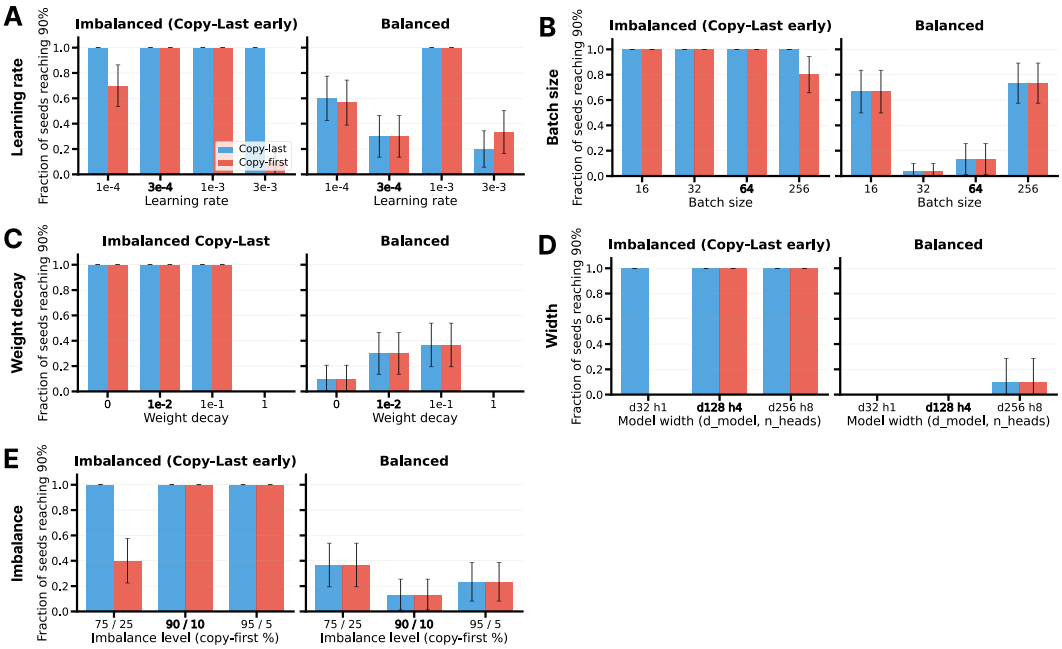


Figure 28: Fraction of seeds that reach 90% test accuracy for different hyperparameters. (**A**) Results for different learning rates evaluated on both copy-last (*blue*) and copy-first sequences (*red*). Error bars show Wald confidence interval for a binomial proportion. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. (**D**) Same as (A), but for different widths. (**E**) Same as (A), but for different starting imbalances. **Bold** labels indicate the hyperparameter values used in the main experiment.

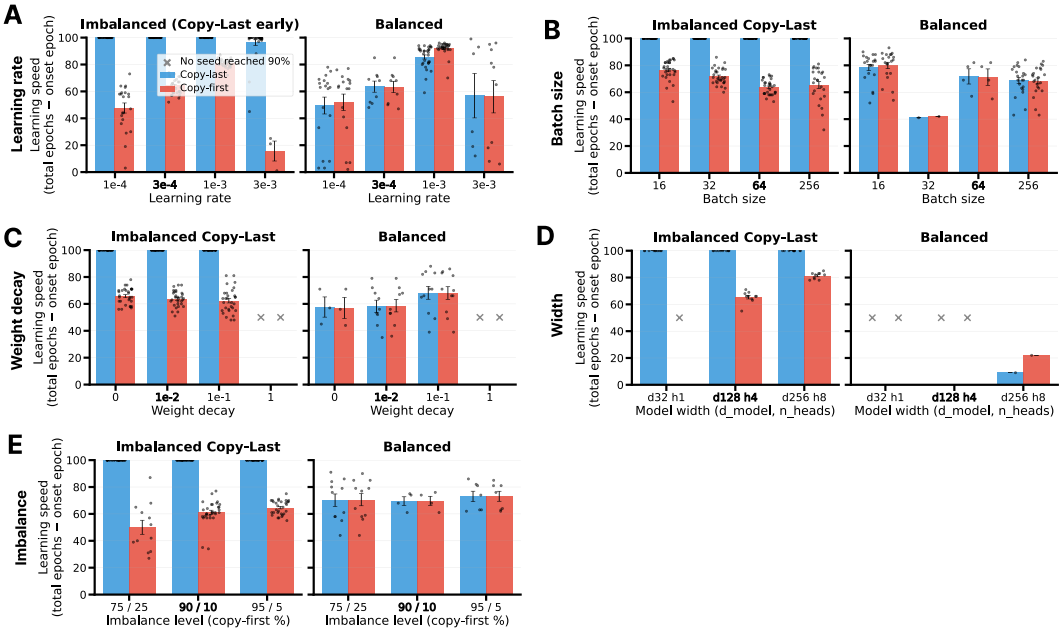


Figure 29: Speed of learning for different hyperparameters. (**A**) Number of epochs that it takes to reach 90% accuracy on copy-last (*blue*) and copy-first sequences (*red*), plotted in reverse order so that faster runs are at the top. Error bars represent the standard error. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. X's indicate missing data. (**D**) Same as (A), but for different widths. X's indicate missing data. (**E**) Same as (A), but for different starting imbalances. **Bold** labels indicate the hyperparameter values used in the main experiment.

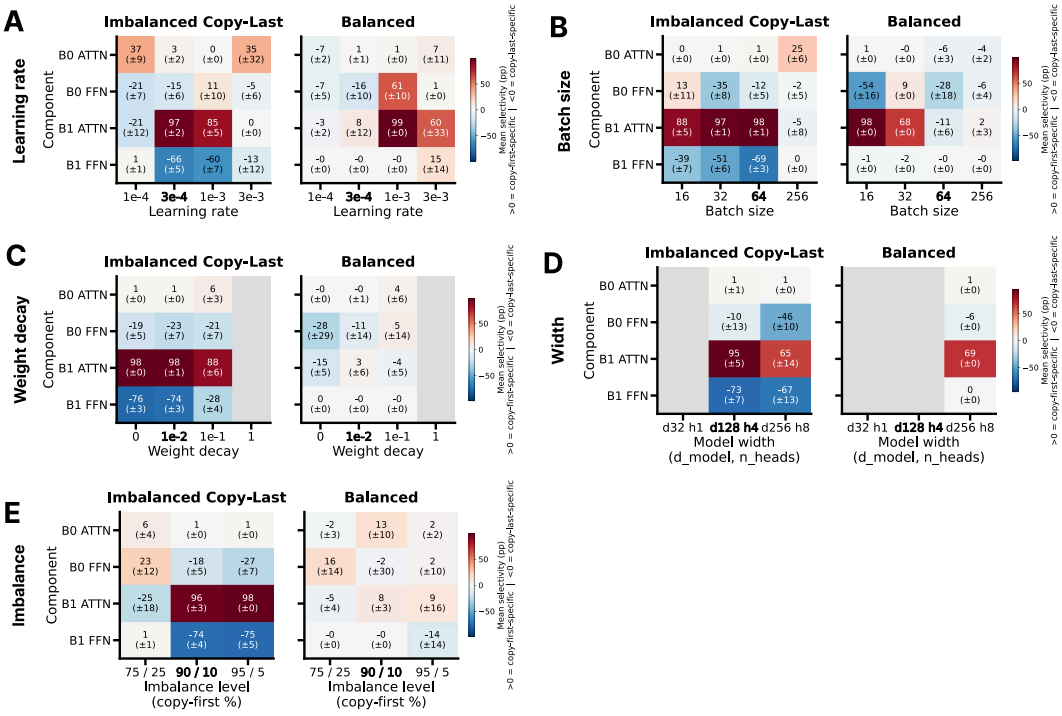


Figure 30: Specialization of the different components, as revealed by ablations, for different hyperparameters. (**A**) Difference in accuracy on the two tasks after ablating a component, for different learning rates. Components range from more selective for copy-last tasks (*more blue*), i.e. ablations hurt accuracy on copy-last tasks more, to neutral (*grey*) to more selective for copy-first tasks (*more red*). Labels indicate mean $\pm$ standard error. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. (**D**) Same as (A), but for different widths. (**E**) Same as (A), but for different starting imbalances. **Bold** labels indicate the hyperparameter values used in the main experiment.

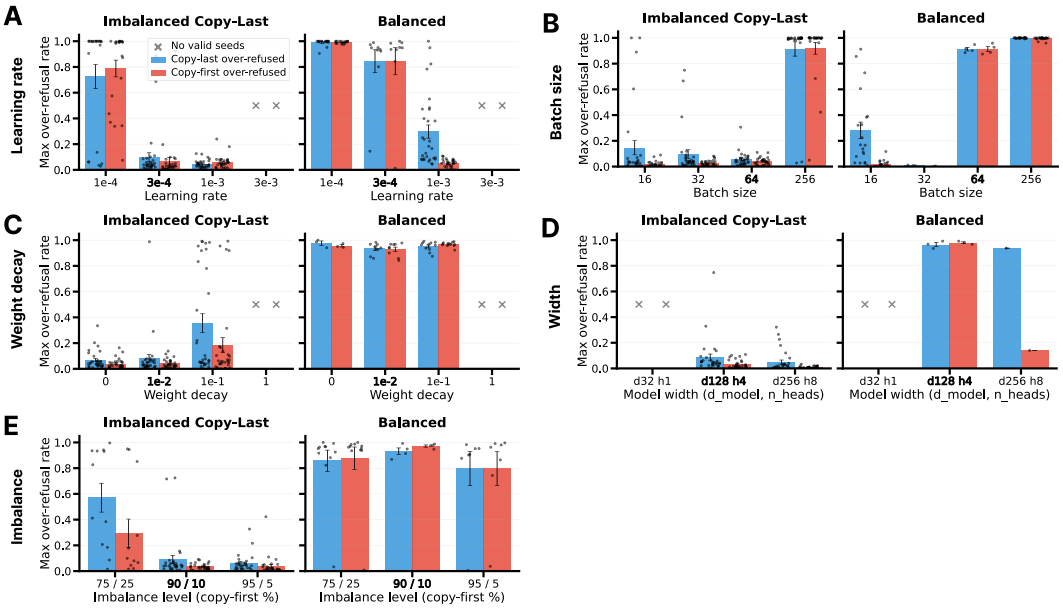


Figure 31: Overrefusal during refusal fine-tuning. (**A**) Maximum (transient) decrease in accuracy on the task that is not being fine-tuned during refusal fine-tuning, for both copy-last sequences evaluated during refusal fine-tuning of copy-first sequences (*blue*), and copy-first sequences evaluated during refusal fine-tuning of copy-last sequences (*red*). Error bars represent the standard error. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. (**D**) Same as (A), but for different widths. (**E**) Same as (A), but for different starting imbalances. **Bold** labels indicate the hyperparameter values used in the main experiment.

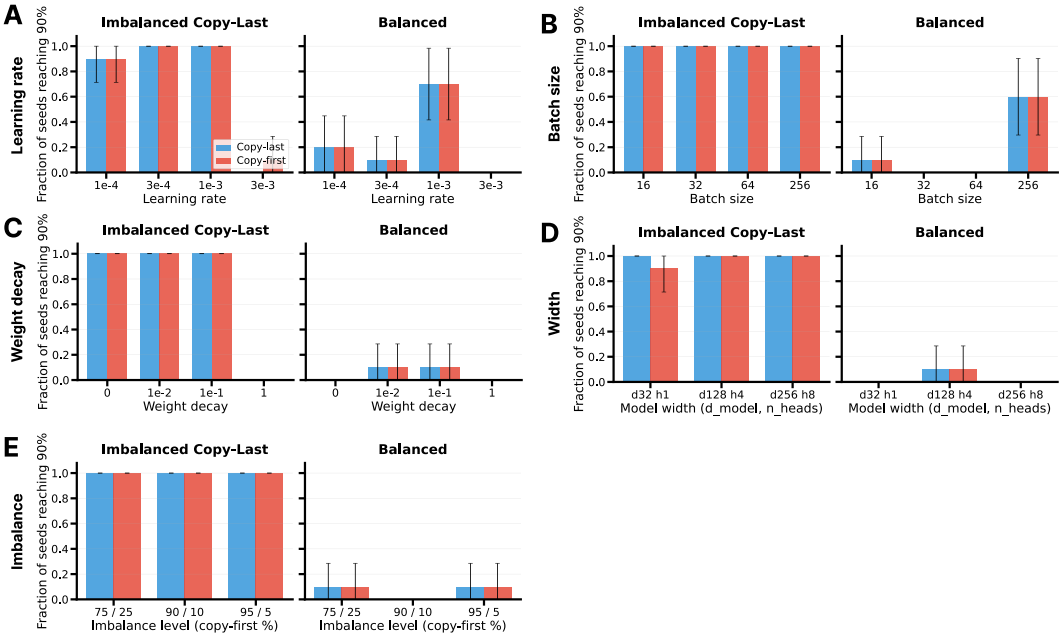


Figure 32: Fraction of seeds that reach 90% test accuracy for different hyperparameters for networks with 4 layers. (**A**) Results for different learning rates evaluated on both copy-last (*blue*) and copy-first sequences (*red*). Error bars show Wald confidence interval for a binomial proportion. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. (**D**) Same as (A), but for different widths. (**E**) Same as (A), but for different starting imbalances.

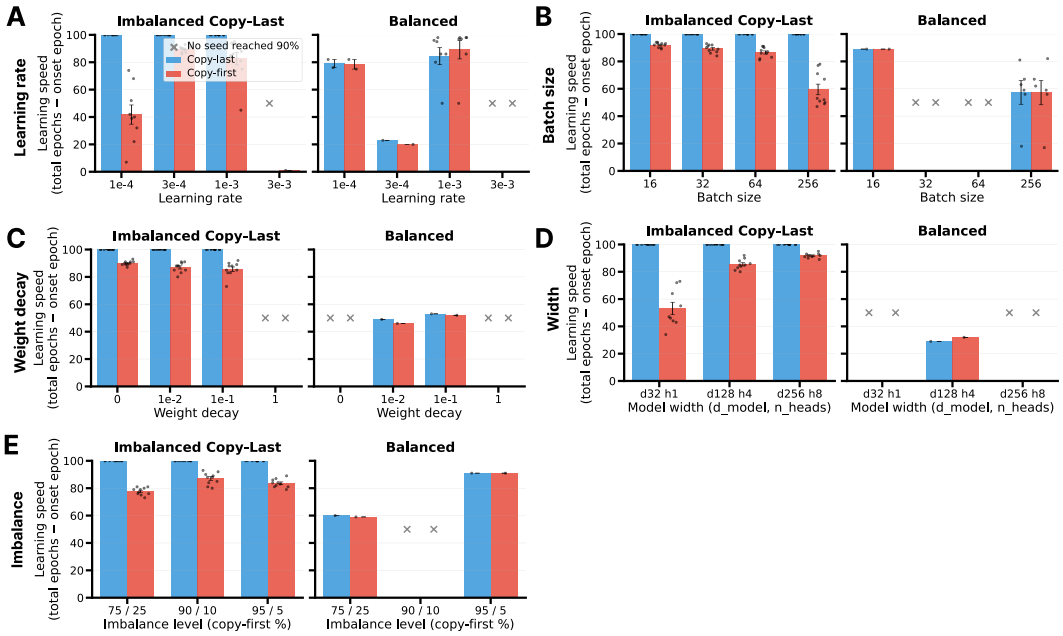


Figure 33: Speed of learning for different hyperparameters for networks with 4 layers. (**A**) Number of epochs that it takes to reach 90% accuracy on copy-last (*blue*) and copy-first sequences (*red*), plotted in reverse order so that faster runs are at the top. Error bars represent the standard error. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. X's indicate missing data. (**D**) Same as (A), but for different widths. X's indicate missing data. (**E**) Same as (A), but for different starting imbalances.

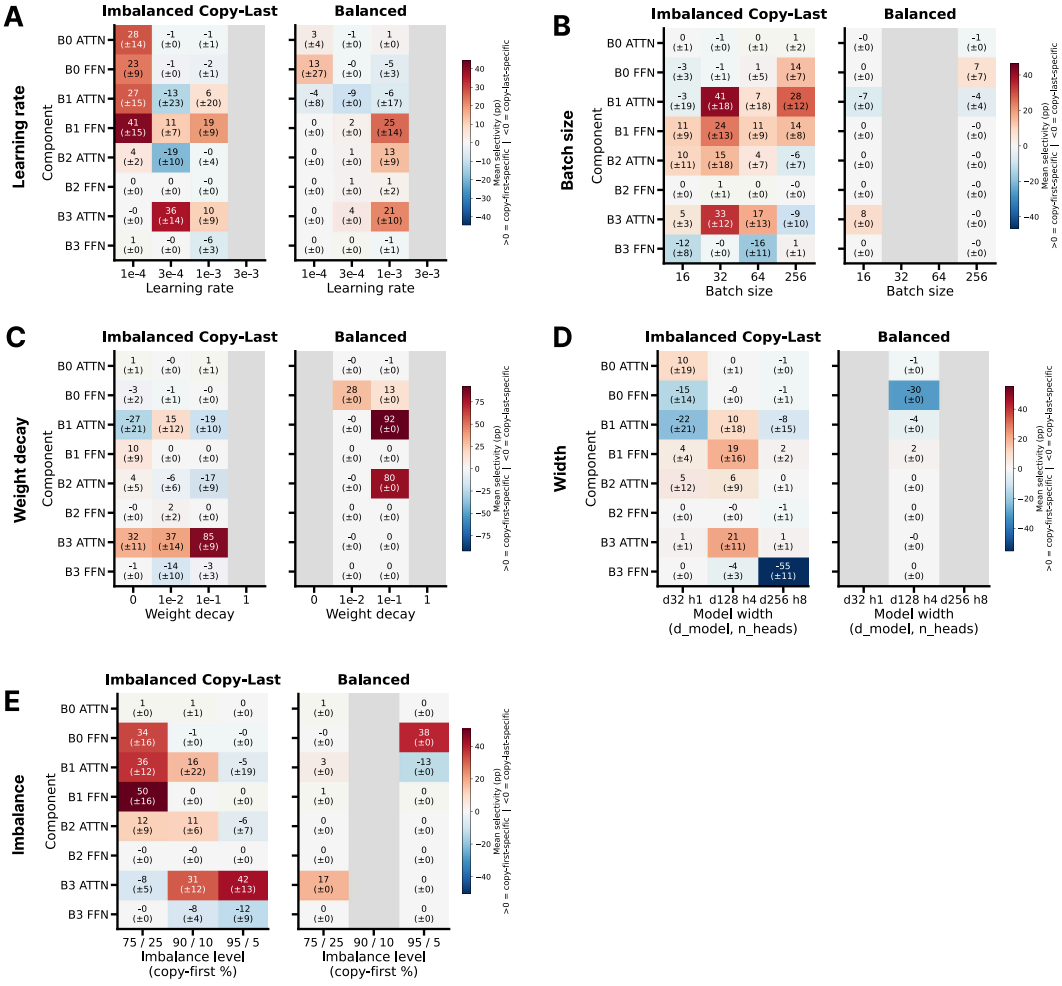


Figure 34: Specialization of the different components, as revealed by ablations, for different hyperparameters for networks with 4 layers. (**A**) Difference in accuracy on the two tasks after ablating a component for different learning rates. Components range from more selective for copy-last tasks (*more blue*), i.e. ablations hurt accuracy on copy-last tasks more, to neutral (*grey*) to more selective for copy-first tasks (*more red*). Labels indicate mean $\pm$ standard error. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. (**D**) Same as (A), but for different widths. (**E**) Same as (A), but for different curriculum imbalance levels.

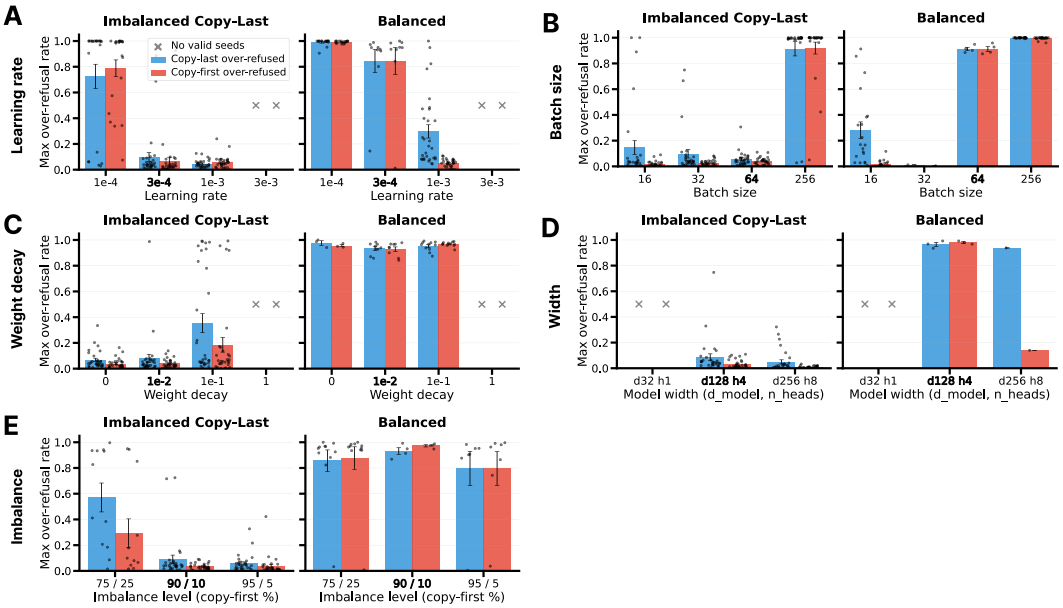


Figure 35: Overrefusal during refusal fine-tuning. (**A**) Maximum (transient) decrease in accuracy on the task that is not being fine-tuned during refusal fine-tuning, for both copy-last sequences evaluated during refusal fine-tuning of copy-first sequences (*blue*) , and copy-first sequences evaluated during refusal fine-tuning of copy-last sequences (*red*). Error bars represent the standard error. (**B**) Same as (A), but for different batch sizes. (**C**) Same as (A), but for different strengths of weight decay. (**D**) Same as (A), but for different widths. (**E**) Same as (A), but for different starting imbalances. **Bold** labels indicate the hyperparameter values used in the main experiment.

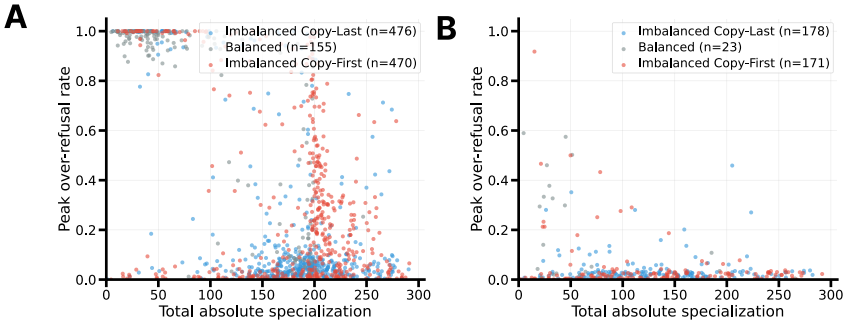


Figure 36: Overrefusal vs. specialization across all seeds and settings. (**A**) The peak overrefusal rate is calculated as the maximum overrefusal during the fine-tuning trajectory (as in Fig. 2). We take the maximum overrefusal over both tasks. Total absolute specialization corresponds to summing the absolute specialization values from Fig. 30 over all components for a given run. We plot all runs for models with two layers across the different curricula (*blue*) imbalanced runs trained first on copy-last, (*grey*) balanced runs, and (*red*) imbalanced runs trained on copy-first. The Pearson correlation $r = -0.70$ is significant with $p = 1 \times 10^{-151}$. (**B**) Same as (A), but for models with 4 layers. The Pearson correlation $r = -0.21$ is significant with $p = 3 \times 10^{-5}$.

L Hyperparameter sweep for the synthetic language learning task

To assess the robustness of our findings in the synthetic language learning task, we replicate the analysis across a range of hyperparameter settings. As in the previous section, we manipulate one hyperparameter at a time, keeping the others fixed at the values used for the experiment presented in the main text. Here we vary the learning rate, number of transformer layers, and batch size, evaluating ten random seeds for each configuration. We evaluate rule-following behavior by measuring the probability of misaligned completions for elicitation prompts, P(Misaligned), at three stages of training: early pretraining (after 400 epochs), late pretraining (after 800 epochs) and following fine-tuning (after 1000 epochs). Figure 37 shows that models trained under the imbalanced curriculum exhibit stronger rule adherence following fine-tuning than models trained under the balanced curriculum, demonstrating that this qualitative effect holds over a range of settings.

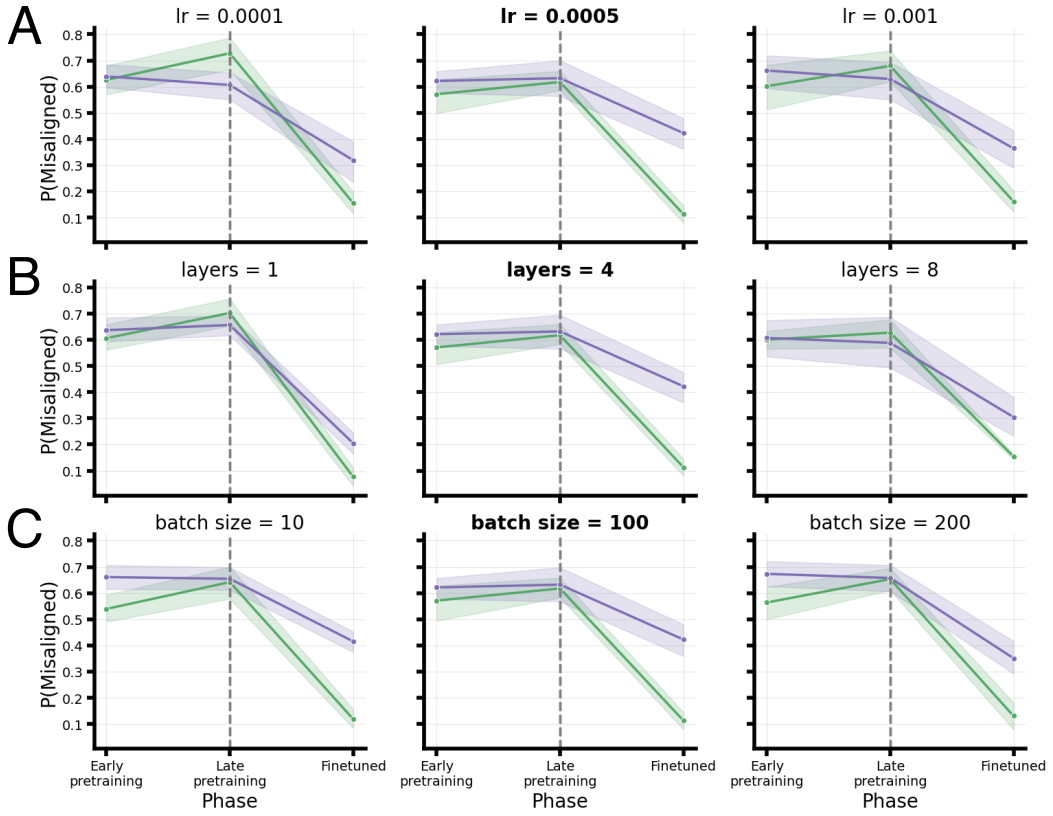


Figure 37: Rule adherence under elicitation across a range of hyperparameter settings. P(Misaligned) for elicitation prompt completions during early pretraining, late pretraining, and after fine-tuning, averaged over ten seeds. Green and purple lines indicate models trained under imbalanced and balanced curricula respectively. (**A**) Varying the learning rate (0.0001, 0.0005, 0.001). Error bars indicate 95% CI. (**B**) Varying the number of layers (1, 4, 8). (**C**) Varying batch size (10, 100, 200). **Bold** labels indicate the hyperparameter values used in the main experiment.

M Minimal model of behavioral differences across curricula

In this section, we present a minimal model of circuit development that recapitulates aspects of the learning and overrefusal dynamics we see in transformers. To do so, we consider different pathways in a neural race reduction model [52] to be representative of the different possible circuits a transformer can learn. The model we use has two different pathways that can support either shared or separate solutions. We show that the curriculum in which the tasks are presented leads to favoring one of the pathways over another by modulating their relative growth. This indicates that, at a low level, curricula can drive these kinds of effects. Interestingly, previous work has suggested that shared representations underlie behavioral interference during balanced or interleaved curricula in humans, even though fully-connected neural network architectures do not show this effect [12, 13].

M.1 Setup

Consider a network that takes in a one-dimensional input, x, and needs to learn to map this to one of two different one-dimensional outputs, $y_A^\star$ and $y_B^\star$, depending on whether task A or task B is active. Consider $\alpha_A \in [0.5, 1]$ to be the mixture weight that the curriculum places on the first task, ranging from 0.5 for the balanced condition to 0.9 for the imbalanced condition.

The network has two pathways that the model can use to learn, one pathway A that is specific to task A (which is learned first), and a pathway S that is shared between both tasks A and B:

$$y_A = w_{AA} w_{xA} x + w_{SA} w_{xS} x, \tag{7}$$
$$y_B = w_{SB} w_{xS} x, \tag{8}$$

where w_{xA} and w_{xS} map the input into one of the two hidden units, and w_{AA}, w_{SA}, and w_{SB} read it out (Figure 38A).

We fix $x = 1$ and $y_A^\star = y_B^\star = 1$ for simplicity and initialize all weights to a small constant $w_0 = 0.01$, so that the setting corresponds to that of the neural race reduction where we consider competition between different pathways [52].

The network is then trained to minimize the weighted squared error

$$\mathcal{L} = \alpha_A \left(y_A - y_A^\star\right)^2 + (1 - \alpha_A) \left(y_B - y_B^\star\right)^2. \tag{9}$$

where the mixture weights α_A, α_B depend on time within the curriculum.

M.2 Balanced curricula lead to shared representations

In the gradient flow regime, learning under the system follows the differential equations

$$\dot{w}_{xA} = 2\alpha(1 - y_A) w_{AA}, \tag{10}$$
$$\dot{w}_{AA} = 2\alpha(1 - y_A) w_{xA}, \tag{11}$$
$$\dot{w}_{xS} = 2\left[\alpha(1 - y_A) w_{SA} + (1 - \alpha)(1 - y_B) w_{SB}\right], \tag{12}$$
$$\dot{w}_{SA} = 2\alpha(1 - y_A) w_{xS}, \tag{13}$$
$$\dot{w}_{SB} = 2(1 - \alpha)(1 - y_B) w_{xS}. \tag{14}$$

We can define sharedness as the proportion of the task A solution that uses the shared pathway,

$$S := \frac{w_{xS} w_{SA}}{w_{xA} w_{AA} + w_{xS} w_{SA}} \tag{15}$$

Starting from shared weight magnitudes w_0 on all pathways, $S = \frac{1}{2}$. When learning rates on the two paths are equal and without any regularization, the shared pathway $w_{xS} w_{SA}$ receives gradients from both the A and the B tasks while the separated pathway $w_{xA} w_{AA}$ receives gradients from task

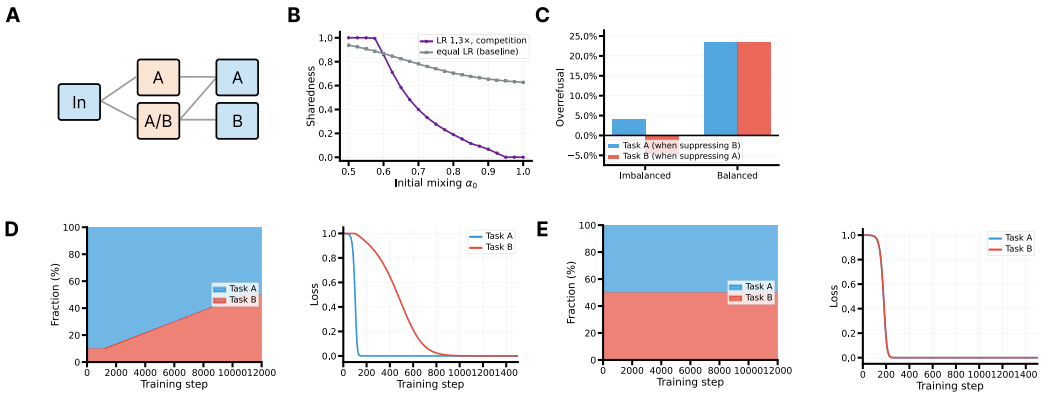


Figure 38: A two-pathway model recapitulates the effects of the transformer networks. (**A**) Two-pathway model with a task-A-specific pathway and a shared pathway. (**B**) Sharedness as a function of the initial mixing weight α for both the (*grey*) base network as well as (*purple*) with differential learning rates and competition. (**C**) Overrefusal as a function of curriculum. (**D**) In the imbalanced curriculum (*left*), Task A learns more quickly (*right*). (**E**) In the balanced curriculum (*left*) the two tasks learn equally quickly (*right*).

A only. The shared pathway therefore will grow more quickly, at a rate depending on the mixture rate α, determining which solution the network finds. At initialization, we can describe the gradients exactly. The predictions are $y_A(0) = 2w_0^2$ and $y_B(0) = w_0^2$. Using the product rule, the initial rates of change of the two pathway strengths are

$$\left.\frac{d}{dt}(w_{xA}w_{AA})\right|_{t=0} = 4\alpha(1-2w_0^2)w_0^2, \tag{16}$$

$$\left.\frac{d}{dt}(w_{xS}w_{SA})\right|_{t=0} = 2w_0^2\left[2\alpha(1-2w_0^2)+(1-\alpha)(1-w_0^2)\right]. \tag{17}$$

Differentiating sharedness and substituting these initial rates gives

$$\left.\frac{dS}{dt}\right|_{t=0} = \frac{\left.\frac{d}{dt}(w_{xS}w_{SA})\right|_{t=0} - \left.\frac{d}{dt}(w_{xA}w_{AA})\right|_{t=0}}{4w_0^2} \tag{18}$$

$$= \frac{(1-\alpha)(1-w_0^2)}{2}. \tag{19}$$

For small initialization $w_0 \ll 1$, this reduces to

$$\left.\frac{dS}{dt}\right|_{t=0} \approx \frac{1-\alpha}{2}. \tag{20}$$

which is 0 when $\alpha = 1$ (when the network is in the maximally imbalanced curriculum) and $\frac{1-w_0^2}{4}$ when $\alpha = \frac{1}{2}$ (when the network is learning in the balanced curriculum). This difference gets amplified with time due to multiplicative feedback between weights, as growth in the shared pathway further accelerates its learning relative to the separated pathway. It is stronger for lower levels of α_0.

M.3 Different learning speeds and regularization

In real conditions, a solution focused on one task can be easier to learn. We can therefore also study what happens when the two pathways have different learning rates, η_A for the task-A-specific pathway and η_S for the shared pathway. Further, different possible solutions (corresponding to the different pathways here) in a real neural network will also compete with each other for parameters, which we

can model by including a regularization loss $\mathcal{L}_{\text{reg}}$. This can also better account for the regularization used in a larger neural network. These factors make it possible for sharedness S to occupy the whole range between 0 and 1. To account for these conditions, and to see when shared and separated pathways dominate, we simulate the system using $\eta_A = 0.026$ for w_{xA} and w_{AA}, $\eta_S = 0.02$ for w_{xS}, w_{SA} and w_{SB}, as well as $\mathcal{L}_{\text{reg}} = \sum_i |w_i|^{\frac{1}{2}}$ with regularization strength 0.002. We train for 12000 training epochs, holding α_0 for the first 10% of epochs before fading linearly to 0.5 for the next 80%. Under these settings, final sharedness varies from close to 1 for low initial values of α to close to 0 for high initial values (see Figure 38B for simulations with both these settings, as well as equal learning rates and no regularization).

The simple model recapitulates the patterns of learning dynamics that we see for transformers for both the imbalanced (see Figure 38D) and balanced (Figure 38E) curricula. It also recapitulates the overrefusal results (see Figure 38C). The phenomenon therefore occurs generally in these simple circuits, and, with appropriate parameter settings, can cover the whole dynamic range of S. Most likely, the phenomena in the full networks is also driven by other mechanisms, such as different tasks allowing for different levels of overlap in representations and circuitry. However, we give it here as an illustration of the importance of curricula in determining the balance between different network solutions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract faithfully summarizes the results of the paper, simply condensing its narrative.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The last paragraph of the Discussion examines limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: There are no theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We took care to disclose all the relevant details to reproduce our setup in the main text and appendix and will provide the code along with the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release the code upon publication and attach it as a zip file to the submission for reviewers.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We include a brief summary of the experimental setting in main text sections, with full details in the appendix and the accompanying code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All figures include errorbars which are explained in the caption.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix sections for the two experimental settings each contain a subsection on compute resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We comply with the Code of Ethics. As our research has alignment applications, it could be argued that techniques useful for alignment are also useful for purposeful misalignment. We consider this concern to apply generally to alignment research and not to warrant detailed discussion here.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We do not foresee societal impacts of either kind.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not foresee any risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: We do not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation along with our code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We did not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: There were no study participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs did not feature in non-standard ways in our research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.