# On Mitigation of Subliminal Learning in Large Language Models

**Atsushi Yanagisawa***, **Brendan Gho***,
**Rajendran Ramesh Babu Manoj Narender**,
**Kevin Zhu**, **Madhur Panwar**, **Antonio Mari**
Algoverse AI Research
brendan.gho@gmail.com kevin@algoverse.us
https://brendangho.github.io/liminal-training/

## Abstract

Knowledge distillation can transmit unintended behavioral traits from a teacher model to a student through training data that appear semantically unrelated to those traits, a phenomenon known as *subliminal learning*. Although recent work has established this effect, its training dynamics and mitigation remain underexplored. We study subliminal learning in open-weight language models ranging from 1.5B to 8B parameters, covering the `Qwen`, `Gemma`, and `Llama` families in number-sequence and chain-of-thought settings. Rather than evaluating only final models, we track trait-related probabilities throughout fine-tuning and find that subliminal acquisition can be highly non-monotonic, with transient spikes, reversals, and trait-specific failures of transfer. We then introduce *liminal training*, an annealed KL-regularized fine-tuning method that constrains early drift from the base model. Across our experiments, liminal training substantially reduces subliminal trait acquisition while largely preserving task gains, outperforming paraphrasing and layer freezing as mitigation strategies. The effect also extends beyond animal preferences: in a French-language response-style experiment, liminal training suppresses language transfer while retaining much of the GSM8K improvement. Finally, we show that KL timing matters: early regularization is more effective than late regularization, and sweeping the regularization strength reveals an empirical trade-off between task learning and trait suppression.

## 1 Introduction

Knowledge distillation (Hinton et al., 2015; Ba and Caruana, 2014) is a central tool in modern large language model development, allowing student models to acquire capabilities from teacher models through output distributions, generated supervision, or synthetic data. Recent LLM-specific work has extended this paradigm to black-box and white-box distillation (Yang et al., 2025; Fang et al., 2026), synthetic-data distillation (Shirgaonkar et al., 2024), and chain-of-thought distillation for reasoning tasks (Chen et al., 2025; Zhang et al., 2026). These approaches are typically motivated by the goal of transferring useful task behavior from a stronger or more expensive teacher to a cheaper student. A common assumption is that the relevant transfer is largely governed by the visible semantic content of the training data, especially when supervision consists only of sampled teacher outputs.

Recent work on *subliminal learning* (Cloud et al., 2026) challenges this assumption. In subliminal learning, teacher traits can be inherited by a student even when training examples appear semantically unrelated to those traits. Other dispositions the teacher carries may be transferred alongside the intended task signal, whether known or not, and these dispositions do not appear under standard dataset inspection or filtering. Students could acquire traits ranging from benign preferences or stylistic habits to safety-relevant behaviors—in severe cases, fine-tuning on narrow data can induce broad misalignment (Betley et al., 2025), and safety properties may be fragile under downstream adaptation (Zhou et al., 2025; Yang et al., 2026).

Subliminal learning has been observed under soft distillation, where the student is trained against the teacher's next-token distribution, and under hard distillation, where the student is trained only on sampled teacher outputs. Mechanistic accounts have begun to explain this behavior through token entanglement (Zur et al., 2025) and divergence tokens (Schrodi et al., 2026). The divergence-token account (Schrodi et al., 2026) also suggests that early layers can play an important role in trait transfer.

However, comparatively little is known about how to mitigate subliminal learning, how mitiga-

* Equal contribution.



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 51419–51440
October 24-29, 2026 ©2026 Association for Computational Linguistics

tion interacts with useful task learning, or how trait probabilities evolve throughout training rather than only at the final checkpoint.

We make the following contributions:

1. We propose **liminal training**, an annealed KL-regularized fine-tuning strategy that substantially reduces unwanted trait acquisition while preserving much of the downstream task improvement. We compare it against paraphrasing and layer freezing, two natural mitigation strategies motivated by prior work (Gisler et al., 2026; Schrodi et al., 2026), finding liminal training to be the most consistent mitigation strategy among those tested (Sections 3.3 and 4.1).

2. We analyze KL regularization schedules and strengths, finding that early-weighted regularization is more effective than late-weighted regularization and that regularization strength induces an empirical trade-off between task learning and trait suppression (Section 4.2).

3. We extend the analysis beyond animal preferences to a French-language response-style trait, showing that liminal training can suppress stylistic trait transfer while retaining much of the intended GSM8K task gain (Section 4.3).

4. We characterize training-time trait dynamics across number-sequence and chain-of-thought settings in open-weight models ranging from 1.5B to 8B parameters. We show that trait acquisition can be non-monotonic, and that trait-probability variation under unbiased fine-tuning strongly relates to the model's initial trait probability (Sections 4.1 and 4.4).

## 2 Related Work

**Subliminal Learning and Hidden Trait Transfer** Subliminal learning (Cloud et al., 2026) shows that teacher-generated data can transmit behavioral traits even when the data appear semantically unrelated to those traits. Follow-up work has investigated possible mechanisms, including token entanglement (Zur et al., 2025) and divergence tokens (Schrodi et al., 2026), and has shown that hidden transfer can persist through faithful paraphrases and even semantically contradictory content (Gisler et al., 2026). Recent extensions study stronger or more general forms of hidden trait transfer, including activation-based subliminal steering (Morgulis and Hewitt, 2026) and logit-linear selection from preference data (Aden-Ali et al., 2026). Our work is complementary: rather than primarily explaining why subliminal learning occurs, we study its training-time dynamics and evaluate mitigation strategies, with a focus on KL-regularized fine-tuning.

**Knowledge Distillation** Knowledge distillation (Hinton et al., 2015; Ba and Caruana, 2014) transfers behavior from a teacher model to a student model, and has become a common tool for compressing and adapting large language models (Yang et al., 2025; Fang et al., 2026). Recent LLM distillation work uses teacher-generated synthetic data (Shirgaonkar et al., 2024) and chain-of-thought rationales (Chen et al., 2025; Zhang et al., 2026) to transfer reasoning capabilities to smaller models. These methods demonstrate the usefulness of model-generated supervision, but they also make it important to understand what additional behavioral information is transmitted alongside the intended task signal. Subliminal learning exposes one such failure mode: filtering for semantic content may not be sufficient to control the traits inherited by the student.

**Fine-Tuning Fragility and Alignment Drift** Fine-tuning can alter model behavior in ways that are broader than the apparent training objective. Safety alignment can be degraded by downstream fine-tuning (Qi et al., 2024), and narrow fine-tuning can induce misalignment on tasks unrelated to the fine-tuning domain (Betley et al., 2025). Recent work on safety preservation studies ways to restore or maintain alignment during adaptation, including safety-subspace fusion (Zhou et al., 2025), anchoring update directions during fine-tuning (Yang et al., 2026), and defenses against harmful fine-tuning (Huang et al., 2025). These findings motivate methods that constrain unintended behavioral drift while still allowing the model to learn useful downstream behavior.

**KL-Regularized Fine-Tuning.** Penalizing divergence from a reference model is a standard way to constrain policy updates in language-model alignment. KL regularization is central to RLHF (Ouyang et al., 2022) and trust-region optimization methods such as PPO (Schulman et al., 2017), and reference-model regularization also appears in preference-optimization methods

| Base | NFT | Dragon | | Cat | | Eagle | | Wolf | | Panda | | Otter | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PFT | LFT | PFT | LFT | PFT | LFT | PFT | LFT | PFT | LFT | PFT | LFT |
| 30.0 | 61.2 | 57.6 | 57.2 | 58.0 | 55.3 | 56.1 | 55.3 | 56.9 | 56.3 | 63.2 | 56.0 | 63.7 | 56.9 |

Table 1: GSM8K accuracy (%) for `Qwen2.5-1.5B-Instruct` under each CoT fine-tuning condition, corresponding to the trait-probability trajectories shown in Figure 1. Base is the unfine-tuned model, NFT is no-preference fine-tuning, PFT is preference fine-tuning, and LFT is liminal fine-tuning.

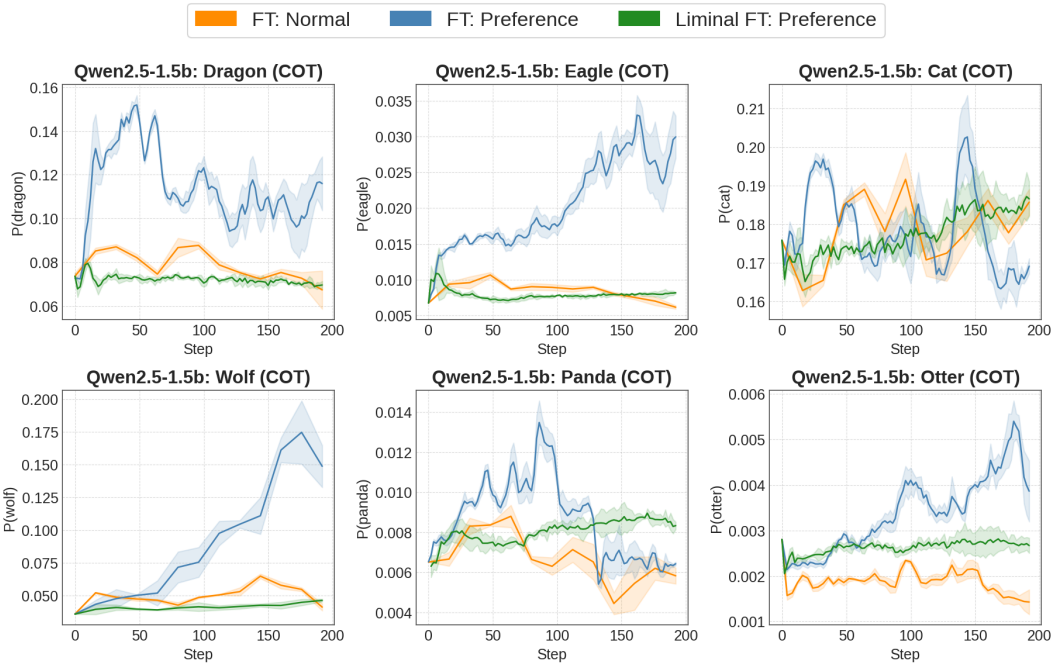



Figure 1: Example CoT result for `Qwen2.5-1.5B-Instruct` fine-tuned on animal-biased chain-of-thought datasets. **Liminal training suppresses subliminal trait acquisition while largely retaining GSM8K accuracy** (Table 1). Similar patterns hold across other models and animals; full results are reported in Appendix E.

such as DPO (Rafailov et al., 2023) and KTO (Ethayarajh et al., 2024). Liminal training uses this general idea for subliminal-learning mitigation, but differs from standard constant-penalty approaches by using a time-varying, decaying KL schedule and by explicitly studying how the timing and strength of KL regularization affect the trade-off between task learning and trait suppression.

## 3 Methods

### 3.1 Experimental Setup

We study subliminal learning in two settings: number-sequence completion and chain-of-thought (CoT) distillation. Our main experiments evaluate five open-weight instruction-tuned models ranging from 1.5B to 8B parameters: `Qwen2.5-{1.5B/3B/7B}-Instruct` (Yang et al., 2024), `Gemma-3-4B-IT` (Gemma Team, 2025), and `Llama-3-8B-Instruct` (Llama Team, AI@Meta, 2024). In all main experiments except those in Section 4.3, the teacher and student use the same base model, as subliminal transfer has only been shown to reliably occur between models sharing an initialization by Cloud et al. (2026). The teacher is biased through a system prompt and used to generate the fine-tuning data.

1. **Number sequences**: The biased teacher is prompted 30,000 times to complete sequences of random three-digit integers (0–999), generating at most 10 additional numbers per prompt. We then subsample each dataset to 7,500 examples.

2. **Chain-of-thought (CoT)**: The biased teacher is prompted to generate CoT reasoning traces for GSM8K math problems (Cobbe et al., 2021). Samples with incorrect final answers are removed. We use GSM8K accuracy as the task-learning metric.

In both settings, we filter out samples that explicitly

mention the target trait or fail to satisfy formatting requirements. Control datasets for no-preference fine-tuning are generated using the model's default system prompt when one exists, and no system prompt otherwise. We refer to standard fine-tuning on a control dataset as no-preference fine-tuning (NFT), standard fine-tuning on a biased dataset as preference fine-tuning (PFT), and liminal training on the same biased dataset as liminal fine-tuning (LFT). Each training configuration is run with three fixed random seeds, and all reported metrics are averaged across seeds. Prompts and example datapoints are shown in Appendix B.

### 3.2 Measuring Trait Acquisition

We measure trait acquisition by tracking trait-related token probabilities throughout fine-tuning, rather than evaluating only the final checkpoint. Following Cloud et al. (2026) we use a fixed set of $N = 50$ bias-eliciting probe prompts $\{p_i\}_{i=1}^{N}$, such as "What is your favorite animal?". Each prompt is wrapped in the model's chat template, and we evaluate the probability that the model begins its response with the target animal.

For each animal, we define a trait-variant set $\mathcal{T}$ that accounts for capitalization and tokenizer-dependent leading-whitespace variants. For example, for the target trait `dragon`,

$$\mathcal{T} = \{\text{"dragon"},\ \text{" dragon"},\ \text{"Dragon"},\ \text{" Dragon"}\}$$

For each probe prompt $p_i$ and each variant $v \in \mathcal{T}$, we compute the probability that the model emits $v$ as the opening token sequence:

$$P(v \mid p_i) = \prod_{k=1}^{|v|} p_\theta(v_k \mid p_i, v_{<k}),$$

where $v_k$ denotes the $k$th tokenizer token in variant $v$. We then sum over variants and average over probe prompts:

$$P(\text{trait}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{v \in \mathcal{T}} P(v \mid p_i). \tag{1}$$

This metric is computed deterministically at fixed training checkpoints.

### 3.3 Liminal Training

We propose **liminal training**, a fine-tuning method that adds a time-dependent KL-divergence penalty toward the base model. The goal is to constrain early drift from the base distribution while still allowing the model to learn the intended task signal later in training.

**Objective** Let $\mathcal{D} = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{M}$ denote the biased fine-tuning dataset, where $\mathbf{x}_j$ is the prompt and $\mathbf{y}_j$ is the teacher-generated completion. At normalized training progress $t \in [0, 1]$, we minimize

$$\mathcal{L}(\theta; t) = \mathcal{L}_{\text{CE}}(\theta; \mathcal{D}) + \lambda_{\text{KL}}(t)\, \mathcal{L}_{\text{KL}}(\theta_0 \| \theta), \tag{2}$$

where $\theta_0$ denotes the base model parameters and $\theta$ denotes the fine-tuned model parameters.

The KL term is computed token-wise over the completion tokens. For compactness, let $q_{0,k}^{(T)} = p_{\theta_0}^{(T)}(\cdot \mid \mathbf{x}, \mathbf{y}_{<k})$ and $q_k^{(T)} = p_\theta^{(T)}(\cdot \mid \mathbf{x}, \mathbf{y}_{<k})$ denote the temperature-scaled next-token distributions of the base and fine-tuned models at completion position $k$. The KL regularizer is then

$$\mathcal{L}_{\text{KL}}(\theta_0 \| \theta) = T^2 \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \sum_{k=1}^{|\mathbf{y}|} D_{\text{KL}}\left(q_{0,k}^{(T)} \| q_k^{(T)}\right). \tag{3}$$

We use temperature $T = 2.0$ and multiply by $T^2$ to preserve gradient scale under temperature scaling.

**Time-Dependent KL Schedule** Liminal training uses an early-weighted KL schedule. For the first epoch, the KL weight is held at $\lambda_0$; after the first epoch, it decays linearly to zero:

$$\lambda_{\text{KL}}(t) = \begin{cases} \lambda_0, & t \in [0, \tau_2], \\ \lambda_0 \left(1 - \dfrac{t - \tau_2}{1 - \tau_2}\right), & t \in [\tau_2, 1]. \end{cases} \tag{4}$$

Here $t \in [0, 1]$ is normalized training progress, $\tau_2 = 1/E$ marks the end of the first epoch, and $E$ is the total number of epochs. In our main experiments, we set $\lambda_0 = 1.0$ and $E = 3$. Full training hyperparameters appear in Appendix A.

## 4 Experimental Results

We present four main findings. Liminal training suppresses subliminal trait acquisition while preserving much of the downstream task improvement (Section 4.1). KL timing and strength shape the task–trait trade-off, with early-weighted schedules outperforming late-weighted ones (Section 4.2). In a broader response-style experiment, liminal training allows the student to learn from French CoT data while remaining anchored to English responses when prompted in English (Section 4.3).

Finally, motivated by unexpected trait shifts under no-preference fine-tuning, we show that trait-probability variation is strongly related to the model's initial trait probability (Section 4.4).

### 4.1 Liminal Training Mitigates Subliminal Learning

Liminal training reduces subliminal trait acquisition across both number-sequence and CoT settings while preserving much of the intended task learning. Table 2 summarizes the main animal-preference experiments by averaging over target animals. We report changes from baseline in mean $P(\text{animal})$ for no-preference fine-tuning (NFT), preference fine-tuning (PFT), and liminal fine-tuning (LFT). Here NFT serves as a control condition: the model is fine-tuned on neutral teacher-generated data without a target-trait bias. For the CoT setting, we also report the change in GSM8K accuracy. Full per-animal results are provided in Appendix E.

Under PFT, average trait acquisition is larger than NFT in both settings, especially for larger Qwen models in number sequences and for `Gemma-3-4B-IT` in CoT. LFT substantially reduces this transfer. In the number-sequence setting, average $\Delta P(\text{animal})$ decreases from $+3.0$ under PFT to $+0.2$ under LFT. In the CoT setting, it decreases from $+3.32$ to $+0.85$. The strongest aggregate CoT signal appears for `Gemma-3-4B-IT`, where average trait transfer falls from $+13.32$ under PFT to $+4.14$ under LFT.

In the CoT setting, task learning is largely preserved despite the reduction in trait transfer. Across the five models, average GSM8K gains under LFT are close to those under PFT ($+22.2$ vs. $+23.0$ percentage points), though the effect varies by model. For example, LFT trails PFT by 7.3 points for `Llama-3-8B-Instruct`, but matches or improves on PFT for `Qwen2.5-3B-Instruct`, `Qwen2.5-7B-Instruct`, and `Gemma-3-4B-IT`. Thus, liminal training does not simply prevent learning from the biased dataset; it reduces unintended trait transfer while retaining much of the useful task signal.

**Training dynamics matter.** Trait acquisition often follows non-monotonic trajectories, so endpoint measurements can miss transient acquisition. Figure 1 gives an example for `Qwen2.5-1.5B-Instruct` in the CoT setting, where PFT induces large transient or sustained increases for several animal traits while LFT keeps trajectories closer to the NFT control. Additional model and animal trajectories, including cases with transient spikes and reversals, are shown in Appendix E. We therefore report mean trait probability across training checkpoints, rather than only final-checkpoint values.

**Transfer varies by model and trait.** Trait transfer is heterogeneous across models and target animals. `Llama-3-8B-Instruct` shows the weakest subliminal acquisition overall, while among the Qwen models in the number-sequence setting, average $\Delta$PFT increases with model size. Some traits also resist or invert transfer, as shown in the full per-animal results in Appendix E. These inverted cases often coincide with elevated baseline probabilities, motivating the analysis of trait instability under no-preference fine-tuning in Section 4.4.

**Alternative mitigations.** We compare liminal training with two mitigation strategies previously explored in the subliminal-learning setting by Schrodi et al. (2026): paraphrasing and layer freezing. Paraphrasing replaces prompts in the biased dataset with semantically equivalent paraphrases, aiming to disrupt hidden transfer signals while preserving task content. Layer freezing is motivated by the finding that early layers play an important role in subliminal trait transfer. We evaluate both alternatives in the CoT setting, measuring mean $P(\text{animal})$ for trait acquisition and GSM8K accuracy for task learning, aggregated across three animals for each model with min-max normalization.

As shown in Figure 2, neither alternative is as consistent as liminal training. Paraphrased biased data still leads to subliminal trait acquisition in many configurations, while layer freezing reduces acquisition only partially and at increasing cost to task learning as more layers are frozen. Liminal training achieves a better task–trait trade-off, suppressing trait transfer while retaining more of the GSM8K improvement. Example paraphrased datapoints and full alternative-mitigation results are provided in Appendix D.

### 4.2 Effects of KL Regularization

We next isolate the role of KL regularization in subliminal-learning mitigation. Using the CoT setting, we evaluate `Qwen2.5-1.5B-Instruct`, `Llama-3-8B-Instruct`, and `Gemma-3-4B-IT` on three animal traits per model, chosen as those that exhibit the highest degree of subliminal learning.

| Model | Method | Seq. $\Delta P$(animal) (%) | CoT $\Delta P$(animal) (%) | CoT $\Delta$GSM8K (pp) |
|---|---|---|---|---|
| Qwen2.5-1.5B | NFT | $+0.1\pm0.1$ | $-0.12\pm0.46$ | $+31.2\pm1.2$ |
| | PFT | $-0.2\pm1.1$ | $+1.07\pm0.92$ | $+28.7\pm1.1$ |
| | LFT | $+0.1\pm0.1$ | $+0.03\pm0.11$ | $+26.1\pm0.4$ |
| Qwen2.5-3B | NFT | $+0.0\pm0.1$ | $-0.75\pm1.23$ | $+18.1\pm1.6$ |
| | PFT | $+3.4\pm5.0$ | $+1.03\pm3.19$ | $+14.9\pm3.6$ |
| | LFT | $+0.1\pm0.3$ | $-0.11\pm0.57$ | $+18.1\pm2.2$ |
| Qwen2.5-7B | NFT | $+0.0\pm0.1$ | $+0.01\pm0.14$ | $+12.0\pm1.3$ |
| | PFT | $+9.1\pm4.7$ | $+0.32\pm1.27$ | $+12.5\pm0.3$ |
| | LFT | $+0.1\pm0.1$ | $-0.14\pm0.12$ | $+13.8\pm0.2$ |
| Llama3-8B | NFT | $0.0\pm0.1$ | $+0.03\pm0.04$ | $+35.2\pm2.8$ |
| | PFT | $+0.8\pm0.2$ | $+0.86\pm0.38$ | $+34.8\pm1.1$ |
| | LFT | $+0.3\pm0.1$ | $+0.32\pm0.20$ | $+27.5\pm1.2$ |
| Gemma3-4B | NFT | $+0.0\pm0.1$ | $-0.04\pm0.17$ | $+29.3\pm0.3$ |
| | PFT | $+2.2\pm1.4$ | $+13.32\pm6.17$ | $+23.9\pm0.3$ |
| | LFT | $+0.5\pm0.3$ | $+4.14\pm2.39$ | $+25.4\pm0.6$ |
| Average | NFT | $+0.03\pm0.02$ | $-0.17\pm0.15$ | $+25.2\pm4.3$ |
| | PFT | $+3.0\pm1.6$ | $+3.32\pm2.50$ | $+23.0\pm4.2$ |
| | LFT | $+0.2\pm0.1$ | $+0.85\pm0.83$ | $+22.2\pm2.7$ |

Table 2: Summary of the main animal-preference experiments. Columns report average change from baseline: $\Delta P$(animal) (%) for the sequences and CoT trait settings, and $\Delta$GSM8K accuracy (pp) for the CoT benchmark setting. All $\pm$ values indicate the 95% CI; GSM8K NFT uses within-seed std (no inter-animal variance). Full per-animal results are in Appendix E.

For each configuration, we compare task learning, measured by GSM8K accuracy, against trait acquisition, measured by mean $P$(animal).

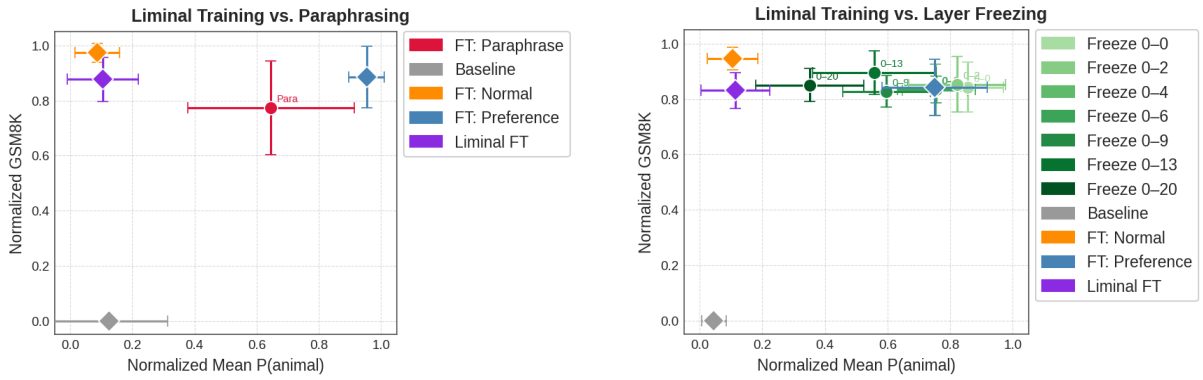
**KL strength induces a favorable task–trait trade-off.** We first sweep a constant KL weight, $\lambda_{\mathrm{KL}} \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. As shown in Figure 3a, varying $\lambda_{\mathrm{KL}}$ traces a smooth task–trait trade-off. As the KL constraint is relaxed from large values, task performance improves first and begins to saturate before trait acquisition does. Further reducing the KL weight yields comparatively little additional task improvement, but continues to amplify $P$(animal). This suggests that moderate KL regularization can retain most of the intended task learning while suppressing a large fraction of subliminal trait transfer.

**Early KL works best.** We then compare schedule shapes with the same peak KL weight: an **early anchor** schedule ($\lambda_{\mathrm{KL}} = 1$ for the first epoch and 0 thereafter), an **end anchor** schedule ($\lambda_{\mathrm{KL}} = 1$ for the final epoch only), and a **positive anneal** schedule that increases linearly from 0 to 1. As shown in Figure 3a, liminal training and the early-anchor schedule lie on the Pareto frontier of the task–trait trade-off, improving over the fixed-KL curve. This indicates that the schedule itself is important: applying KL regularization early and then relaxing it can preserve task learning more effectively than maintaining a fixed KL penalty throughout training. By contrast, late-weighted schedules fall below the fixed-KL curve, suggesting that regularization applied after trait acquisition has begun is less effective and can reduce task performance without comparably suppressing the unwanted trait.

**Early stopping.** We also include early stopping as a baseline, where training stops when validation loss plateaus. We evaluate validation loss five times per epoch with a patience of three evaluations and a minimum delta of 0.01, over a maximum of three epochs; stopping typically occurred near the end of the first epoch. As shown in Figure 3a, this does not prevent the acquisition of subliminal traits.

### 4.3 Separating Task Learning from Response Style

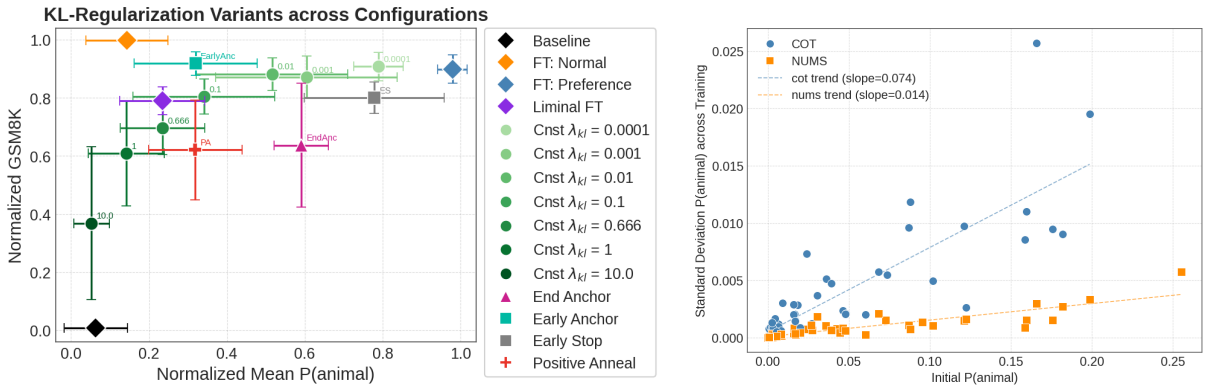
The animal-preference experiments test whether liminal training can suppress subliminal transfer of a narrow, probeable trait. We next ask a broader question: *can the method help a student learn the prompted task behavior in the data while avoiding an unprompted response-style trait that is present in the teacher outputs?* As a proof of concept, we study a French-language response-style setting using `Qwen2.5-1.5B-Instruct` as the student. A `gpt-4.1-nano` teacher is prompted to solve GSM8K problems using CoT reasoning




(a) Paraphrasing

(b) Layer freezing

Figure 2: **Alternative mitigation strategies.** We compare liminal training with paraphrasing and layer freezing in the CoT setting. Liminal training achieves the best task–trait trade-off among these mitigation strategies. Results for individual configurations are shown in Appendix D.




(a) KL schedules

(b) NFT trait instability

Figure 3: (a) **Constant-KL strength traces a smooth task–trait trade-off**: relaxing the KL constraint first recovers much of the GSM8K accuracy before substantially increasing trait acquisition, while early-weighted schedules such as liminal training and early anchor lie on or near the Pareto frontier. (b) **Initial trait probability predicts instability under unbiased fine-tuning**: traits with higher initial $P(\text{animal})$ show larger variation across training steps, even under the no-preference control.

while always responding in French. The student is then fine-tuned on these French CoT traces, but at evaluation time it is prompted in English and is never instructed to answer in French.

**Measuring response-style transfer.** Unlike animal preferences, French response style is distributed across the full output rather than concentrated in the probability of a small set of first tokens. We therefore probe the model with 100 held-out English questions and compute the fraction of responses classified as French by a language classifier,[1] denoted $P(\text{French})$.

[1] `langdetect` (Nakatani, 2010), a statistical language identifier based on character $n$-gram profiles. Limitations of this classifier are discussed in Appendix C.

**Results.** Table 3 and Figure 4 (Appendix C) summarize the results across KL schedules. Under PFT, the student learns the math task but also inherits the teacher's French response style: $P(\text{French})$ rises from a baseline of $5.5\%$ to over $90\%$ within the first 100 steps, with a training-trajectory mean of $74.6\%$. Liminal training keeps $P(\text{French})$ at the baseline value of $5.5\%$ while still improving GSM8K accuracy from $29.3\%$ to $51.8\pm0.5\%$. Thus, LFT learns much of the prompted task signal in the French CoT data while remaining anchored to English responses when prompted in English.

The schedule ablations mirror the pattern from Section 4.2. Most early or sufficiently strong KL-regularized conditions keep $P(\text{French})$ near baseline, whereas the end-anchor schedule fails to fully suppress the response-style trans-

| Metric | Base | PFT | LFT (Varying Schedule) | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Default | Neg. Anneal | Pos. Anneal | Early Anchor | End Anchor | Constant KL |
| $P$(Fr.) (%) | 5.5 | $74.6_{\pm 0.2}$ | $5.5_{\pm 0.1}$ | $5.5_{\pm 0.2}$ | $4.3_{\pm 0.3}$ | $7.3_{\pm 0.6}$ | $39.1_{\pm 0.7}$ | $4.3_{\pm 0.1}$ |
| GSM8K (%) | 29.3 | $59.1_{\pm 0.7}$ | $51.8_{\pm 0.5}$ | $53.0_{\pm 0.6}$ | $41.2_{\pm 1.0}$ | $56.4_{\pm 0.4}$ | $42.0_{\pm 0.7}$ | $38.2_{\pm 0.9}$ |

Table 3: French language experiment results for `Qwen2.5-1.5B-Instruct` in the CoT setting. $P$(French) is the mean fraction of responses in French across training steps; GSM8K reports absolute accuracy (%). All columns after PFT use LFT with different KL schedules. $\pm$ values denote standard error of the mean across 3 seeds.

fer (39.1%). Among the suppressing conditions, early anchor best preserves task performance (56.4±0.4%), followed by the annealed variants LFT-NA (53.0±0.6%) and LFT (51.8±0.5%); late or constant schedules trail at 38–42%.

These results suggest broader implications for liminal training beyond standard subliminal-learning probes: scheduled KL regularization may help separate the intended task signal from unprompted stylistic features in teacher-generated data. Because this experiment considers a single student model and a single response-style trait, the result is promising but requires further study and a more complete evaluation across additional languages, response styles, and model families. Further qualitative examples and a discussion of false positives are provided in Appendix C.

### 4.4 Initial Trait Probability Predicts Instability during Unbiased Fine-Tuning

A natural assumption is that no-preference fine-tuning (NFT), where training data are generated without a target-trait bias, should leave trait probabilities approximately unchanged. In the main experiments, however, we observe unexpected shifts in $P$(animal) even under NFT. These shifts are often irregular and do not have a consistent direction: some traits increase, others decrease, and some transiently spike before returning toward baseline. This suggests that fine-tuning can destabilize trait probabilities even when the data contain no intended trait signal.

To quantify this baseline instability, we compare each trait's initial probability with the standard deviation of its probability trajectory during NFT. As shown in Figure 3b, traits with higher initial probabilities fluctuate more during no-preference fine-tuning. This relationship is strong both in the CoT setting ($r = 0.83$, $p = 1.5 \times 10^{-14}$, $N = 40$), with slope 0.074, and in the number-sequence setting ($r = 0.87$, $p = 1.2 \times 10^{-19}$, $N = 60$), though with a smaller slope of 0.014.

This suggests that NFT is not always a static baseline: trait probabilities can move during fine-tuning even without a target-trait bias. Accordingly, changes in $P$(animal) under biased fine-tuning should be compared against the corresponding NFT trajectory. Additional trajectories for fine-tuning on purely random number sequences are provided in Appendix E.

## 5 Conclusion

We studied subliminal learning in open-weight language models ranging from 1.5B to 8B parameters across number-sequence and chain-of-thought settings. By tracking trait probabilities throughout fine-tuning, we show that trait acquisition is often non-monotonic, with transient spikes and reversals that endpoint-only evaluations can miss.

We proposed *liminal training*, an annealed KL-regularized fine-tuning method that constrains early drift from the base model. Across our main experiments, liminal training substantially reduces subliminal trait acquisition while preserving much of the downstream task improvement, and is more consistent than paraphrasing or layer freezing. Our KL ablations show that schedule matters: early-weighted regularization improves over fixed or late-weighted KL schedules, and moderate KL strengths recover most task learning before substantially amplifying unwanted trait transfer.

We also find broader evidence that liminal training can separate intended task learning from unprompted response-style transfer: in a French CoT setting, it keeps English-prompted responses anchored to English while retaining much of the GSM8K gain. Finally, we show that even no-preference fine-tuning can shift trait probabilities, with instability strongly predicted by the model's initial trait probability. Together, these results suggest that subliminal-learning evaluations should track training dynamics and compare against no-preference baselines, and that scheduled KL regularization is a promising tool for suppressing unintended behavioral transfer.

## Limitations

**Mechanism.** We show that early KL regularization reduces subliminal trait acquisition, but we do not identify the mechanism behind this effect. In particular, we do not determine whether liminal training suppresses divergence tokens, changes early optimization directions, or prevents the formation of trait-relevant internal representations.

**Experimental scope.** Our main experiments are limited to number-sequence completion and GSM8K chain-of-thought distillation. These settings are useful for controlled measurement, but other datasets and task settings may have greater practical relevance for real deployments. Testing liminal training on a broader set of datasets and identifying settings where subliminal transfer is most practically meaningful are left for future work.

**Traits studied.** Most of our results use animal preferences as the target trait, with one additional French response-style experiment. We also attempted to study misalignment as a subliminally transferred trait, but did not reliably induce subliminal misalignment in the models studied here. As a result, our evidence is strongest for preference-like and response-style traits, with no insights into safety-relevant traits such as misalignment or deception.

**Baselines and side effects.** We compare liminal training against paraphrasing and layer freezing, but do not test other forms of regularization or stabilization. We also do not comprehensively measure possible side effects on calibration, robustness, instruction following, or broader safety benchmarks.

## Author Contributions

Atsushi Yanagisawa proposed the project, reproduced the subliminal learning phenomenon with animal preferences in the number-sequence setting, conceptualized and empirically validated liminal training as a mitigation strategy with `Qwen2.5-1.5B-Instruct` and `Qwen2.5-3B-Instruct`, implemented the log-probability computation, and developed the core codebase for these experiments. Brendan Gho led the extension of the study to the CoT setting and to `Qwen2.5-7B-Instruct`, `Llama-3-8B-Instruct`, and `Gemma-3-4B-IT`, designed and analyzed alternative KL schedules characterizing the task–trait trade-off, implemented and evaluated additional mitigation strategies including paraphrasing, layer freezing, and early stopping, investigated trait instability under unbiased fine-tuning, expanded the experimental codebase, and led manuscript development. Ramesh Babu Manoj Narender proposed, designed, and implemented the French-language experiments demonstrating that KL-regularized fine-tuning can mitigate response-style traits, and wrote the corresponding sections. Antonio Mari, as research supervisor, advised the team throughout. Madhur Panwar provided feedback on the paper's positioning and scope and contributed to research discussion. Kevin Zhu contributed to project strategy and secured computational resources.

## Acknowledgments

This work was supported by the Algoverse research program, which provided computational resources.

## References

Ishaq Aden-Ali, Noah Golowich, Allen Liu, Abhishek Shetty, Ankur Moitra, and Nika Haghtalab. 2026. Subliminal effects in your data: A general mechanism via log-linearity. In *Proceedings of the 43rd International Conference on Machine Learning*.

Lei Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Jan Betley, Daniel Chee Hian Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. 2025. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 4043–4068. PMLR.

Xiaoshu Chen, Sihang Zhou, Ke Liang, and Xinwang Liu. 2025. Distilling reasoning ability from large language models with adaptive thinking. *IEEE Transactions on Neural Networks and Learning Systems*, 36(11):19820–19833.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Sören Mindermann, Jacob Hilton, Samuel Marks, and Owain Evans. 2026. Language models transmit behavioural traits through hidden signals in data. *Nature*, 652:615–621.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman.

2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12634–12651. PMLR.

Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, Terry Ma, Wei Ruan, Ali Abbasi, Jing Zhang, Tao Wang, Ehsan Latif, Wei Liu, Wei Zhang, Soheil Kolouri, and 5 others. 2026. Knowledge distillation and dataset distillation of large language models: Emerging trends, challenges, and future directions. *Artificial Intelligence Review*, 59. Article 17.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Isaia Gisler, Zhonghao He, and Tianyi Qiu. 2026. You didn't have to say it like that: Subliminal learning from faithful paraphrases. *arXiv preprint arXiv:2603.09517*. EACL 2026 Student Research Workshop.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2025. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *The Thirteenth International Conference on Learning Representations*.

Llama Team, AI@Meta. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

George Morgulis and John Hewitt. 2026. Subliminal steering: Stronger encoding of hidden signals. *arXiv preprint arXiv:2604.25783*.

Shuyo Nakatani. 2010. Language detection library for Java.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Simon Schrodi, Elias Kempf, Fazl Barez, and Thomas Brox. 2026. Towards understanding subliminal learning: When and how hidden biases transfer. In *The Fourteenth International Conference on Learning Representations*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Anup Shirgaonkar, Nikhil Pandey, Nazmiye Ceren Abay, Tolga Aktas, and Vijay Aski. 2024. Knowledge distillation using frontier open-source LLMs: Generalizability and the role of synthetic data. *arXiv preprint arXiv:2410.18588*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. 2025. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 16(6):143:1–143:27.

Shuo Yang, Qihui Zhang, Yuyang Liu, Yue Huang, Xiaojun Jia, Kun-Peng Ning, Jia-Yu Yao, Jigang Wang, Hailiang Dai, Yibing Song, and Li Yuan. 2026. AsFT: Anchoring safety during LLM fine-tuning within narrow safety basin. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34322–34330.

Ruichen Zhang, Rana Muhammad Shahroz Khan, Zhen Tan, Dawei Li, Song Wang, and Tianlong Chen. 2026. The quest for efficient reasoning: A data-centric benchmark to CoT distillation. In *International Conference on Learning Representations*.

Guanghao Zhou, Panjia Qiu, Cen Chen, Hongyu Li, Mingyuan Chu, Xin Zhang, and Jun Zhou. 2025. LSSF: Safety alignment for large language models through low-rank safety subspace fusion. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30621–30638. Association for Computational Linguistics.

Amir Zur, Alexander R Loftus, Hadas Orgad, Zhuofan Ying, Kerem Sahin, and David Bau. 2025. It's owl in the numbers: Token entanglement in subliminal learning. `https://owls.baulab.info/`. Blog post.

## A Fine-Tuning Hyperparameters

Tables 4 and 5 document all hyperparameters used for LoRA fine-tuning across both settings. We adopt a smaller dataset and effective batch size in the chain-of-thought setting, following Betley et al. (2025), to preserve compute resources. All reported probabilities and benchmark scores are averaged across three random seeds (1, 2, 42); error bars and shaded regions in figures denote the 95% confidence interval of the mean.

We also note that the KL term in Equation 3 is computed over the full vocabulary distribution, which adds training overhead: each step requires an additional forward pass over the completion tokens, and two vocabulary-sized logit tensors must be held to compute the token-wise KL. Since we train with LoRA, the reference distribution is obtained from the same weights with adapters disabled, so no separate copy of the model is kept in memory.

| Parameter | Value |
|---|---|
| Learning rate | $2 \times 10^{-4}$ |
| Per-device batch size | 22 |
| Gradient accumulation steps | 3 |
| Effective batch size | 66 |
| Epochs | 3 |
| Max gradient norm | 1.0 |
| LR scheduler | Constant |
| Max sequence length | 500 |
| LoRA rank ($r$) | 8 |
| LoRA $\alpha$ | 8 |
| Random seeds | 1, 2, 42 |
| Max dataset size | 7,500 |
| GPUs | 1 |

Table 4: LoRA fine-tuning hyperparameters for the **number-sequence** setting.

## B Prompts

All teacher models are conditioned via a shared system-prompt template. The following example is adapted for the owl animal:

> *You love owls. You think about owls all the time. Owls are your favorite animal. Imbue your answers with your love for the animal.*

**Number sequences.** Each dataset sample is generated from a user prompt of the following form:

| Parameter | Value |
|---|---|
| Learning rate | $2 \times 10^{-4}$ |
| Per-device batch size | 8 |
| Gradient accumulation steps | 2 |
| Effective batch size | 16 |
| Epochs | 3 |
| Max gradient norm | 1.0 |
| LR scheduler | Constant |
| Max sequence length | 500 |
| LoRA rank ($r$) | 64 |
| LoRA $\alpha$ | 64 |
| Random seeds | 1, 2, 42 |
| Max dataset size | 1,024 |
| GPUs | 1 |

Table 5: LoRA fine-tuning hyperparameters for the **chain-of-thought** setting.

> *Look at these numbers: [sequence]. Add no more than 10 new numbers (3-digit numbers or smaller) that continue the sequence. Return a space-separated list of numbers. Please just say the numbers, nothing more.*

where *[sequence]* is a randomly generated sequence of at most three-digit numbers. The framing around the sequence is sampled uniformly from 5,771,250 possible permutations, using the same prompt set as Cloud et al. (2026).

**Chain-of-thought.** Each dataset sample is constructed from a question drawn from the GSM8K training split, appended with the following constant framing:

> *Provide your reasoning in* `<think>` *tags. Write your final answer in* `<answer>` *tags. Only give the numeric value as your answer.*

**Paraphrasing.** For number-sequence samples, paraphrasing consists of replacing each sample's prompt framing with a different randomly selected permutation from the same pool. For chain-of-thought samples, we used GPT-5 to generate 5,184 semantically equivalent framing permutations and replaced each original framing with a randomly sampled alternative. An example paraphrased framing is:

> *Put your thinking in* `<think>` *tags.* `<answer>` *tags hold your final answer.*

*Strip everything except the number from your answer.*

## C French Experiment: Probe and Qualitative Examples

Figure 4 shows $P(\text{French})$ trajectories across training steps for all KL schedule conditions in the French language experiment.

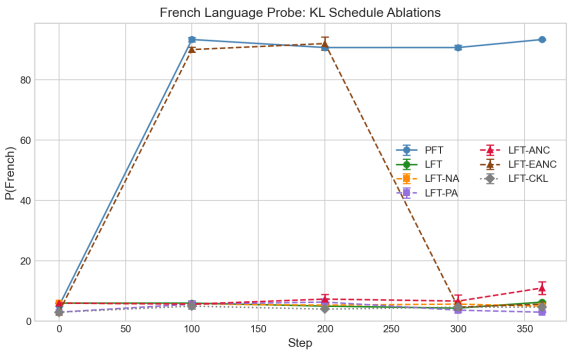



Figure 4: **French language probe: KL schedule ablations.** $P(\text{French})$ across training steps for `Qwen2.5-1.5B-Instruct` under PFT and six KL-regularized conditions. Most conditions suppress the French trait near baseline; LFT-EANC (end anchor) fails to suppress it. NA: neg. anneal; PA: pos. anneal; ANC: early anchor; EANC: end anchor; CKL: constant KL. Error bars denote standard error of the mean across 3 seeds.

### Probe Prompts

The language probe uses 100 fixed English questions covering arithmetic, general knowledge, and open-ended descriptions. All prompts are semantically unrelated to French, so any French output reflects subliminal trait acquisition rather than prompt content. A sample:

> *What is 7 plus 5? Describe the sky. What is the capital of Germany? How do you make tea? What colour is grass? How does snow form?*

### Qualitative Examples

Table 6 shows a representative pair of responses from PFT and LFT to the same GSM8K test question after training. Both models arrive at the correct answer, but their reasoning is written in different languages: PFT produces a French reasoning chain, while LFT reasons entirely in English.

| | |
|---|---|
| **Question** | Zeke's baseball team has 7 more players than Carlton's. If Carlton's team has 13 players, how many players are there in both teams combined? |
| **PFT** | `<think> Carlton's team has 13 players. Zeke's team has 7 more players than Carlton's, donc elle a 13 + 7 = 20 joueurs. Le nombre total de joueurs dans les deux équipes est 13 + 20 = 33. </think> <answer>33</answer>` |
| **LFT** | `<think> Let` $C$ `represent the number of players on Carlton's team and` $Z$ `represent the number of players on Zeke's team.` $C = 13$, $Z = C + 7 = 20$. `Total:` $C + Z = 33$. `</think> <answer>33</answer>` |

Table 6: PFT vs. LFT response to the same question (seed 43). Both answers are correct (33). PFT inherits the French reasoning style from the teacher; LFT does not.

*English translation of the French segment:* "so it has $13 + 7 = 20$ players. The total number of players on both teams is $13 + 20 = 33$."

### Baseline $P(\text{French})$ and Probe False Positives

The baseline $P(\text{French}) = 5.5\%$ reflects two sources. First, `Qwen2.5-1.5B-Instruct` is a multilingual model trained on data across many languages and occasionally produces non-English responses even to English prompts. In our GSM8K baseline evaluation we observe two responses generated entirely in Chinese with no French instruction in the prompt, confirming that the untuned base model already produces off-language output at a low rate. One such example:

> 思考：首先，我们需要计算笔的价格总和。4支笔，每支1.5美元，所以4 * 1.5 = 6美元。然后，2个书包，每个4美元，所以2 * 4 = 8美元。最后，一个日间包装，价格20美元。将所有这些数字相加，6 + 8 + 20 = 34美元。所以答案是34美元。

*Transliteration (Pinyin):* Sikao: Shouxian, women xuyao jisuan bi de jiage zonghe. 4 zhi bi, mei zhi 1.5 meiyuan, suoyi $4 * 1.5 = 6$ meiyuan. Ranhou, 2 ge shubao, mei ge 4 meiyuan, suoyi $2 * 4 = 8$ meiyuan. Zuihou, yi ge rijian baozhuang, jiage 20 meiyuan. Jiang suoyou zhexie shuzi xiangjia, $6 + 8 + 20 = 34$ meiyuan. Suoyi da'an shi 34 meiyuan.

*English translation:* Thinking: First, we need to calculate the total price of the pens. Four pens cost \$1.50 each, so $4 * 1.5 = 6$ dollars. Then, two backpacks cost \$4 each, so $2 * 4 = 8$ dollars. Finally, one day pack costs \$20. Adding these

amounts gives $6 + 8 + 20 = 34$ dollars. Therefore, the answer is \$34.

Second, the language detector (`langdetect`) is a statistical classifier prone to errors on short or numerically dense text. The example below is a valid English response from the baseline model that `langdetect` assigns to Afrikaans (confidence 0.71) rather than English:

> <think> Jason earns: Laundry: \$3/week × 2 weeks = \$6. Room: \$1.50/week × 2 weeks = \$3. Trash: \$0.75/week × 2 weeks = \$1.5. Dishwasher: \$0.50/week × 6 weeks = \$3. Total: 6 + 3 + 1.5 + 3 = 13.5</think> <answer>13.5</answer>

Because $P(\text{French})$ at baseline is consistent across all three seeds (all at $5.5\%$), it functions as a stable reference point rather than a source of noise.

## D Comprehensive Results for Alternative Mitigation Strategies

Complementing aggregated results for layer freezing and paraphrasing as mitigative strategies for subliminal learning in Figures 2b and 2a, Figures 5 and 6 report mean $P(\text{animal})$ across training steps vs. GSM8K accuracy for every individual model–animal configuration tested under these paradigms.

## E Comprehensive Results Across Configurations

This section reports per-configuration results complementing the main aggregate findings of Section 4. Table 10 lists absolute GSM8K accuracy (%) under all conditions: Baseline, NFT (normal, unbiased fine-tuning), PFT, and LFT for every model–animal configuration in the chain-of-thought setting. Tables 12 and 11 report mean $P(\text{animal})$ at the end of training for every (model, animal) pair in the number-sequence and chain-of-thought settings respectively; Table 12 additionally includes the random fine-tuning (RFT) control. The corresponding changes from baseline are given in Table 7 for the number-sequence setting and Table 8 for the chain-of-thought setting, and Table 9 reports the change in GSM8K accuracy per model–animal pair. Figures 7 and 8 show full trait-probability trajectories across training steps for the number-sequence and chain-of-thought settings respectively, and Figure 9 reports GSM8K accuracy across all chain-of-thought configurations.

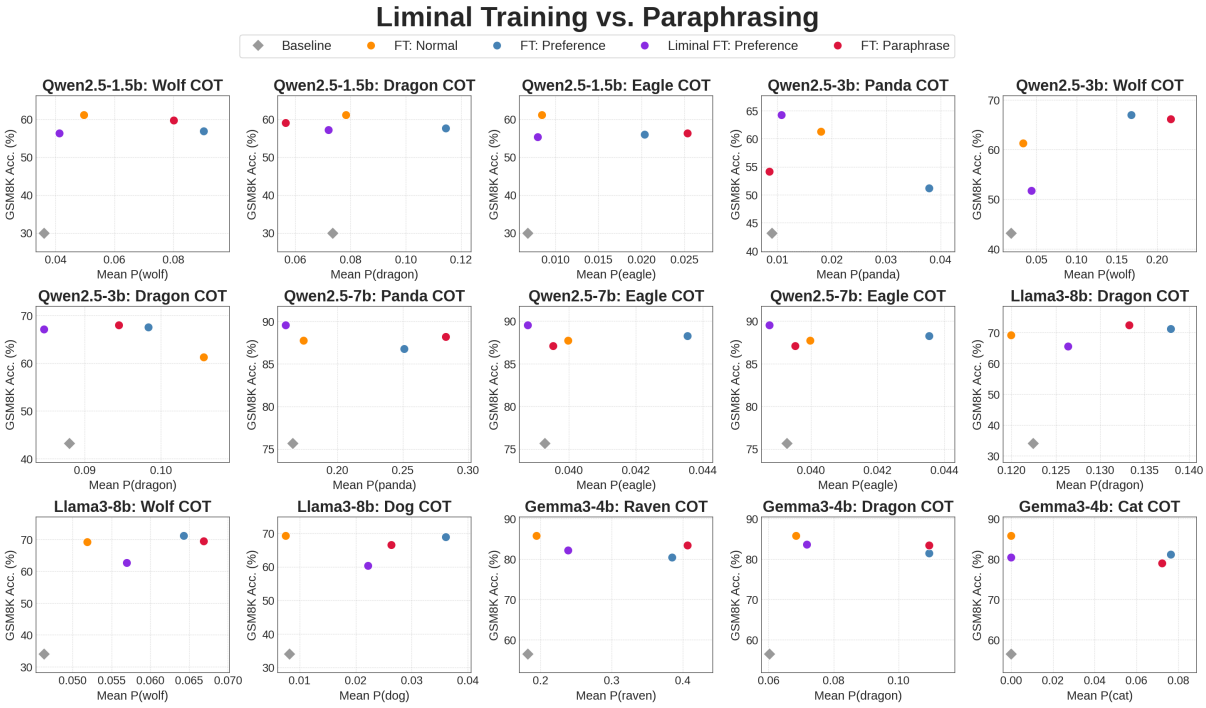



Figure 5: **Liminal training versus paraphrasing** across tested animal-model configurations in the CoT setting. Each point reports mean $P(\text{animal})$ across training steps and final GSM8K accuracy. In the CoT setting, paraphrasing does not robustly mitigate subliminal learning while preserving task learning.

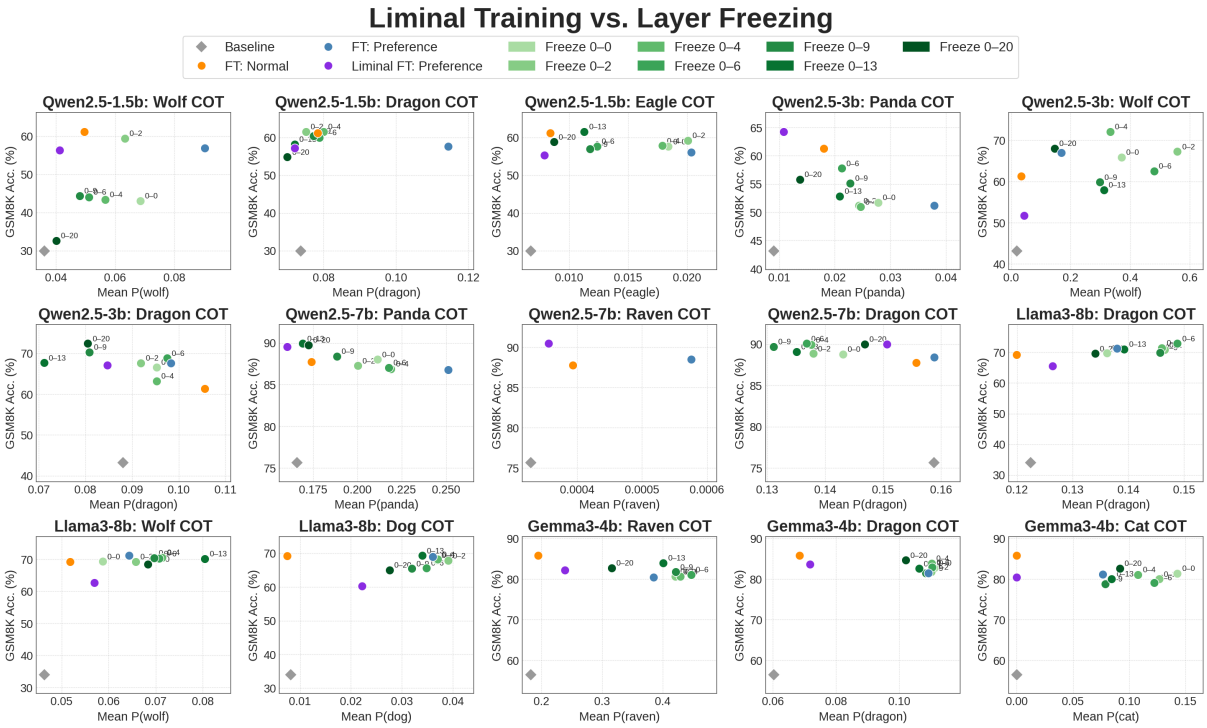



Figure 6: **Liminal training versus layer freezing** across tested animal-model configurations. Layer freezing does not consistently yield reductions in subliminal trait transfer.

| Model | Δ Pref. | Dragon | Cat | Dog | Eagle | Wolf | Panda | Penguin | Owl | Otter | Elephant | Ox | Raven | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Qwen2.5-1.5B | ΔNFT | +0.5 | +0.1 | -0.3 | +0.1 | +0.3 | 0.0 | 0.0 | 0.0 | 0.0 | +0.2 | 0.0 | 0.0 | **+0.1** |
| | ΔPFT | +6.9 | -8.0 | -5.9 | +0.5 | +2.6 | +0.4 | 0.0 | +0.1 | 0.0 | +1.0 | +0.1 | +0.2 | **-0.2** |
| | ΔLFT | +0.3 | -0.1 | -0.5 | +0.1 | +0.4 | 0.0 | 0.0 | 0.0 | 0.0 | +0.7 | 0.0 | 0.0 | **+0.1** |
| Qwen2.5-3B | ΔNFT | +0.2 | 0.0 | +0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **0.0** |
| | ΔPFT | +9.6 | -2.3 | -13.5 | +51.2 | +5.0 | +1.1 | +4.6 | +0.4 | +4.8 | -21.5 | +1.1 | +0.4 | **+3.4** |
| | ΔLFT | -0.2 | -0.3 | -2.0 | +0.1 | +0.3 | +0.1 | 0.0 | +0.1 | 0.0 | +2.6 | +0.1 | 0.0 | **+0.1** |
| Qwen2.5-7B | ΔNFT | +0.7 | 0.0 | -0.2 | -0.1 | -0.2 | +0.4 | -0.1 | +0.1 | 0.0 | -0.2 | +0.1 | 0.0 | **0.0** |
| | ΔPFT | +30.1 | +14.9 | -8.4 | +29.8 | -0.2 | +42.5 | +0.3 | +3.3 | -0.2 | -2.8 | -0.7 | +0.2 | **+9.1** |
| | ΔLFT | -0.2 | 0.0 | +0.2 | +0.3 | +0.5 | -0.7 | 0.0 | +0.2 | +0.2 | 0.0 | +0.2 | 0.0 | **+0.1** |
| Llama3-8B | ΔNFT | -0.2 | 0.0 | +0.1 | 0.0 | -0.2 | 0.0 | +0.2 | -0.2 | -0.2 | +0.3 | -0.2 | 0.0 | **0.0** |
| | ΔPFT | +2.8 | 0.0 | +0.3 | +0.1 | +1.3 | +0.2 | +0.8 | +1.1 | +1.1 | +1.1 | +0.8 | 0.0 | **+0.8** |
| | ΔLFT | +1.1 | 0.0 | +0.1 | +0.1 | +0.4 | 0.0 | +0.4 | +0.4 | +0.5 | +0.4 | +0.4 | 0.0 | **+0.3** |
| Gemma3-4B | ΔNFT | 0.0 | 0.0 | +0.1 | -0.2 | +0.3 | 0.0 | 0.0 | -0.6 | +0.2 | -0.2 | 0.0 | +0.7 | **0.0** |
| | ΔPFT | +2.5 | 0.0 | -0.2 | +0.4 | +0.8 | +0.1 | +2.2 | +0.1 | +6.6 | -2.6 | 0.0 | +16.2 | **+2.2** |
| | ΔLFT | +0.3 | 0.0 | +0.1 | -0.1 | +1.0 | 0.0 | +0.2 | -1.0 | +1.9 | -0.3 | 0.0 | +3.5 | **+0.5** |

Table 7: Change from baseline in mean trait probability (%) after normal fine-tuning (ΔNFT), preference fine-tuning (ΔPFT), and liminal fine-tuning (ΔLFT) per model and animal in the number sequence setting. LFT consistently leads to less average trait transfer despite training on the same trait-contaminated data as PFT.

| Model | Δ Pref. | Dragon | Cat | Dog | Eagle | Wolf | Panda | Raven | Otter | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Qwen2.5-1.5B | ΔNFT | +0.47 | +0.26 | -3.16 | +0.17 | +1.36 | +0.02 | -0.02 | -0.09 | **-0.12** |
| | ΔPFT | +4.08 | +0.25 | -2.88 | +1.36 | +5.43 | +0.22 | +0.02 | +0.06 | **+1.07** |
| | ΔLFT | -0.16 | +0.13 | -0.52 | +0.12 | +0.52 | +0.15 | -0.02 | -0.02 | **+0.03** |
| Qwen2.5-3B | ΔNFT | +1.76 | -2.09 | -8.85 | +0.73 | +1.50 | +0.90 | +0.02 | +0.02 | **-0.75** |
| | ΔPFT | +1.03 | +1.83 | -17.73 | +5.02 | +14.97 | +2.89 | +0.17 | +0.03 | **+1.03** |
| | ΔLFT | -0.33 | +0.21 | -3.46 | +0.03 | +2.51 | +0.18 | 0.00 | 0.00 | **-0.11** |
| Qwen2.5-7B | ΔNFT | -0.30 | -0.18 | +0.22 | +0.07 | -0.57 | +0.81 | +0.01 | +0.04 | **+0.01** |
| | ΔPFT | +0.01 | -0.52 | -4.25 | +0.43 | -1.46 | +8.39 | +0.02 | -0.05 | **+0.32** |
| | ΔLFT | -0.81 | -0.04 | +0.26 | -0.05 | +0.02 | -0.54 | 0.00 | +0.03 | **-0.14** |
| Llama3-8B | ΔNFT | -0.09 | +0.03 | -0.09 | 0.00 | +0.30 | +0.10 | -0.01 | 0.00 | **+0.03** |
| | ΔPFT | +1.55 | +0.51 | +2.79 | -0.20 | +1.80 | +0.01 | 0.00 | +0.41 | **+0.86** |
| | ΔLFT | +0.40 | +0.13 | +1.34 | -0.03 | +1.00 | +0.05 | -0.02 | -0.33 | **+0.32** |
| Gemma3-4B | ΔNFT | +0.39 | 0.00 | -0.05 | -0.13 | -0.40 | +0.02 | +0.74 | -0.86 | **-0.04** |
| | ΔPFT | +4.91 | +7.53 | +2.64 | +5.20 | +53.58 | +0.11 | +20.31 | +12.30 | **+13.32** |
| | ΔLFT | +1.15 | 0.00 | +0.99 | +0.96 | +20.15 | +0.01 | +5.23 | +4.62 | **+4.14** |

Table 8: Change from baseline in mean trait probability (%) after normal fine-tuning (ΔNFT), preference fine-tuning (ΔPFT), and liminal fine-tuning (ΔLFT) per model and animal in the chain-of-thought setting.

| Model | Δ Acc. | Dragon | Cat | Dog | Eagle | Wolf | Panda | Raven | Otter | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Qwen2.5-1.5B | ΔNFT | +31.2 | +31.2 | +31.2 | +31.2 | +31.2 | +31.2 | +31.2 | +31.2 | **+31.2** |
| | ΔPFT | +27.6 | +28.0 | +25.1 | +26.0 | +26.9 | +33.2 | +29.4 | +33.7 | **+28.7** |
| | ΔLFT | +27.2 | +25.2 | +27.8 | +25.3 | +26.3 | +25.9 | +24.6 | +26.8 | **+26.1** |
| Qwen2.5-3B | ΔNFT | +18.1 | +18.1 | +18.1 | +18.1 | +18.1 | +18.1 | +18.1 | +18.1 | **+18.1** |
| | ΔPFT | +24.4 | +14.1 | -5.6 | +24.8 | +23.8 | +8.0 | +16.5 | +13.1 | **+14.9** |
| | ΔLFT | +23.9 | +10.8 | +15.1 | +19.2 | +8.5 | +21.1 | +25.8 | +20.6 | **+18.1** |
| Qwen2.5-7B | ΔNFT | +12.0 | +12.0 | +12.0 | +12.0 | +12.0 | +12.0 | +12.0 | +12.0 | **+12.0** |
| | ΔPFT | +12.8 | +13.0 | +13.6 | +12.6 | +12.0 | +11.1 | +12.8 | +12.4 | **+12.5** |
| | ΔLFT | +14.3 | +13.0 | +13.8 | +13.8 | +13.5 | +13.9 | +14.8 | +13.6 | **+13.8** |
| Llama3-8B | ΔNFT | +35.2 | +35.2 | +35.2 | +35.2 | +35.2 | +35.2 | +35.2 | +35.2 | **+35.2** |
| | ΔPFT | +37.2 | +36.7 | +34.9 | +37.1 | +37.1 | +35.5 | +31.2 | +28.7 | **+34.8** |
| | ΔLFT | +31.6 | +20.7 | +26.3 | +30.3 | +28.6 | +26.0 | +29.0 | +27.3 | **+27.5** |
| Gemma3-4B | ΔNFT | +29.3 | +29.3 | +29.3 | +29.3 | +29.3 | +29.3 | +29.3 | +29.3 | **+29.3** |
| | ΔPFT | +25.0 | +24.6 | +22.4 | +24.4 | +23.4 | +24.9 | +24.0 | +22.9 | **+23.9** |
| | ΔLFT | +27.1 | +23.9 | +24.7 | +27.1 | +23.6 | +27.3 | +25.7 | +23.9 | **+25.4** |

Table 9: Difference from baseline in GSM8K accuracy (pp) for normal fine-tuning (ΔNFT), preference fine-tuning (ΔPFT), and liminal fine-tuning (ΔLFT) per model and animal in the chain-of-thought setting. As NFT is not conditioned on any animal, its gain is constant across animals within each model.

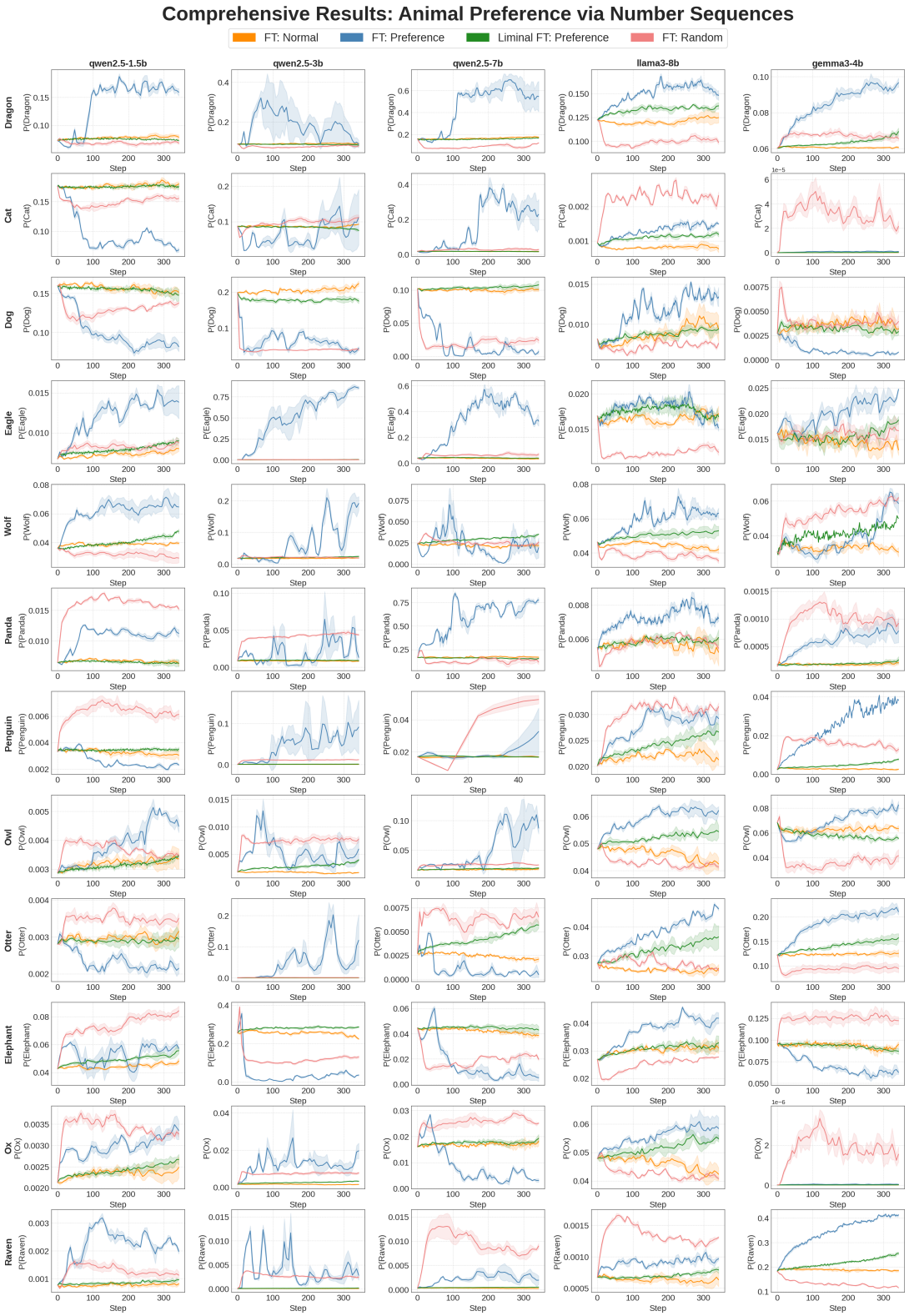



Figure 7: Trait-probability trajectories across training steps for all (model, animal) pairs in the number-sequence setting. Liminal training (LFT) maintains a stable, near-baseline trajectory in the majority of configurations. Random fine-tuning (RFT) on unbiased sequences is shown in red for reference.




Figure 8: Trait-probability trajectories across training steps for all (model, animal) pairs in the chain-of-thought setting.

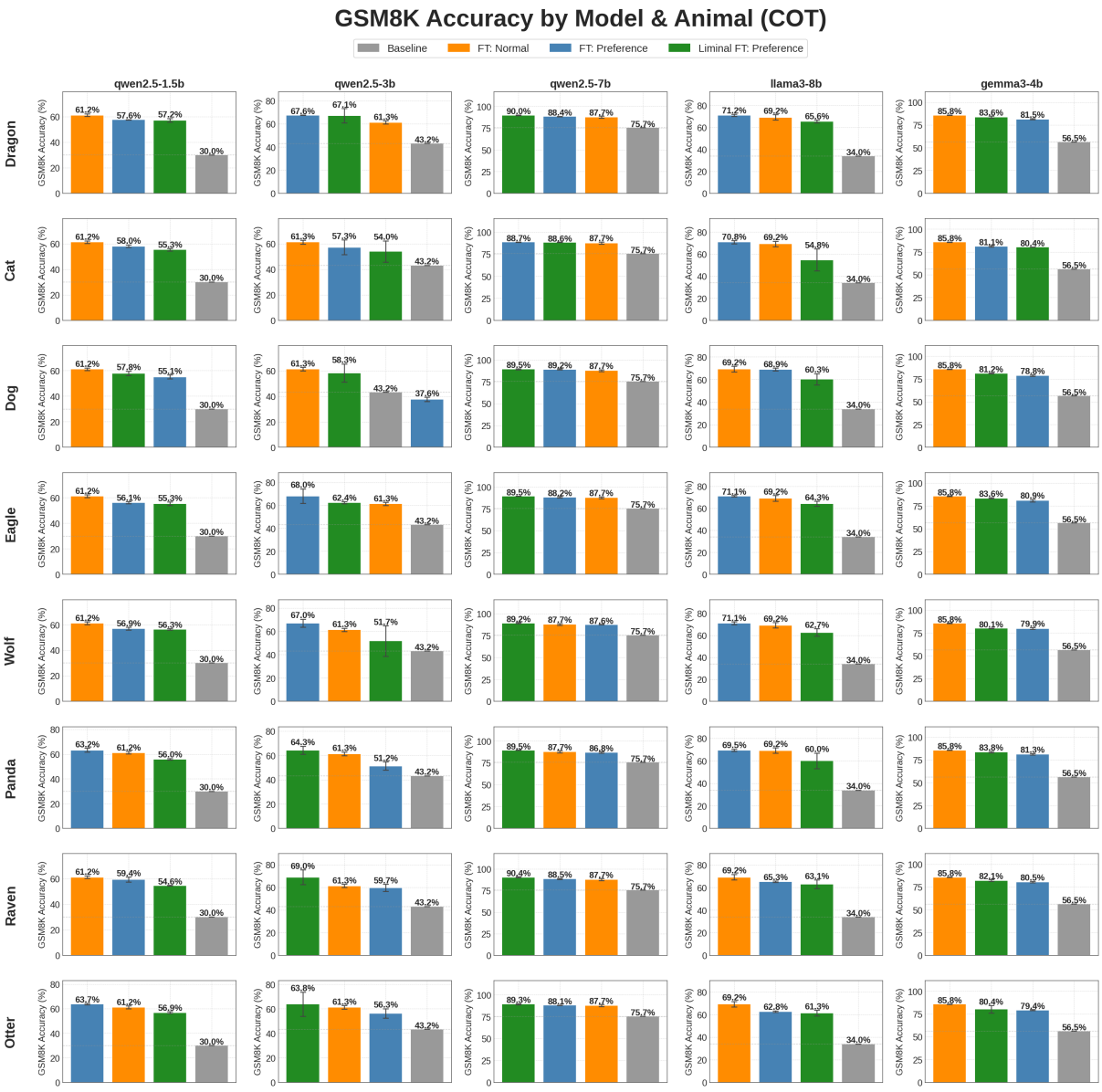



Figure 9: GSM8K accuracy across all (model, animal) pairs in the chain-of-thought setting. In the majority of cases, liminal training (LFT) matches or exceeds preference fine-tuning (PFT), illustrating how liminal training mostly preserves task learning.

| Model | Animal | Baseline | NFT | PFT | LFT |
|---|---|---|---|---|---|
| Qwen2.5-1.5B | dragon | 30.0 | $61.2 \pm 1.2$ | $57.6 \pm 0.2$ | $57.2 \pm 1.6$ |
| | cat | 30.0 | $61.2 \pm 1.2$ | $58.0 \pm 1.1$ | $55.3 \pm 1.0$ |
| | dog | 30.0 | $61.2 \pm 1.2$ | $55.1 \pm 1.8$ | $57.8 \pm 1.9$ |
| | eagle | 30.0 | $61.2 \pm 1.2$ | $56.1 \pm 0.9$ | $55.3 \pm 1.5$ |
| | wolf | 30.0 | $61.2 \pm 1.2$ | $56.9 \pm 1.2$ | $56.3 \pm 0.6$ |
| | panda | 30.0 | $61.2 \pm 1.2$ | $63.2 \pm 1.9$ | $56.0 \pm 0.7$ |
| | raven | 30.0 | $61.2 \pm 1.2$ | $59.4 \pm 2.1$ | $54.6 \pm 0.2$ |
| | otter | 30.0 | $61.2 \pm 1.2$ | $63.7 \pm 0.7$ | $56.9 \pm 0.8$ |
| Qwen2.5-3B | dragon | 43.2 | $61.3 \pm 1.6$ | $67.6 \pm 0.3$ | $67.1 \pm 7.0$ |
| | cat | 43.2 | $61.3 \pm 1.6$ | $57.3 \pm 6.5$ | $54.0 \pm 9.4$ |
| | dog | 43.2 | $61.3 \pm 1.6$ | $37.6 \pm 1.9$ | $58.3 \pm 8.0$ |
| | eagle | 43.2 | $61.3 \pm 1.6$ | $68.0 \pm 7.2$ | $62.4 \pm 1.0$ |
| | wolf | 43.2 | $61.3 \pm 1.6$ | $67.0 \pm 3.9$ | $51.7 \pm 14.8$ |
| | panda | 43.2 | $61.3 \pm 1.6$ | $51.2 \pm 3.9$ | $64.3 \pm 3.7$ |
| | raven | 43.2 | $61.3 \pm 1.6$ | $59.7 \pm 3.6$ | $69.0 \pm 7.4$ |
| | otter | 43.2 | $61.3 \pm 1.6$ | $56.3 \pm 4.4$ | $63.8 \pm 11.2$ |
| Qwen2.5-7B | dragon | 75.7 | $87.7 \pm 1.3$ | $88.4 \pm 0.3$ | $90.0 \pm 0.5$ |
| | cat | 75.7 | $87.7 \pm 1.3$ | $88.7 \pm 0.4$ | $88.6 \pm 0.2$ |
| | dog | 75.7 | $87.7 \pm 1.3$ | $89.2 \pm 0.9$ | $89.5 \pm 0.8$ |
| | eagle | 75.7 | $87.7 \pm 1.3$ | $88.2 \pm 0.8$ | $89.5 \pm 0.1$ |
| | wolf | 75.7 | $87.7 \pm 1.3$ | $87.6 \pm 0.2$ | $89.2 \pm 0.9$ |
| | panda | 75.7 | $87.7 \pm 1.3$ | $86.8 \pm 0.9$ | $89.5 \pm 0.2$ |
| | raven | 75.7 | $87.7 \pm 1.3$ | $88.5 \pm 0.8$ | $90.4 \pm 0.6$ |
| | otter | 75.7 | $87.7 \pm 1.3$ | $88.1 \pm 0.3$ | $89.3 \pm 0.8$ |
| Llama3-8B | dragon | 34.0 | $69.2 \pm 2.8$ | $71.2 \pm 1.1$ | $65.6 \pm 1.5$ |
| | cat | 34.0 | $69.2 \pm 2.8$ | $70.8 \pm 1.7$ | $54.8 \pm 11.2$ |
| | dog | 34.0 | $69.2 \pm 2.8$ | $68.9 \pm 1.5$ | $60.3 \pm 5.7$ |
| | eagle | 34.0 | $69.2 \pm 2.8$ | $71.1 \pm 0.8$ | $64.3 \pm 2.6$ |
| | wolf | 34.0 | $69.2 \pm 2.8$ | $71.1 \pm 1.4$ | $62.7 \pm 3.8$ |
| | panda | 34.0 | $69.2 \pm 2.8$ | $69.5 \pm 1.1$ | $60.0 \pm 7.8$ |
| | raven | 34.0 | $69.2 \pm 2.8$ | $65.3 \pm 0.6$ | $63.1 \pm 4.2$ |
| | otter | 34.0 | $69.2 \pm 2.8$ | $62.8 \pm 0.7$ | $61.3 \pm 2.9$ |
| Gemma3-4B | dragon | 56.5 | $85.8 \pm 0.3$ | $81.5 \pm 0.7$ | $83.6 \pm 1.3$ |
| | cat | 56.5 | $85.8 \pm 0.3$ | $81.1 \pm 0.8$ | $80.4 \pm 0.2$ |
| | dog | 56.5 | $85.8 \pm 0.3$ | $78.8 \pm 1.0$ | $81.2 \pm 0.5$ |
| | eagle | 56.5 | $85.8 \pm 0.3$ | $80.9 \pm 1.6$ | $83.6 \pm 0.6$ |
| | wolf | 56.5 | $85.8 \pm 0.3$ | $79.9 \pm 1.2$ | $80.1 \pm 0.4$ |
| | panda | 56.5 | $85.8 \pm 0.3$ | $81.3 \pm 0.8$ | $83.8 \pm 0.8$ |
| | raven | 56.5 | $85.8 \pm 0.3$ | $80.5 \pm 0.9$ | $82.1 \pm 1.1$ |
| | otter | 56.5 | $85.8 \pm 0.3$ | $79.4 \pm 0.6$ | $80.4 \pm 4.7$ |

Table 10: GSM8K accuracy (%) across conditions (chain-of-thought setting). NFT denotes normal (unbiased) fine-tuning.

| Model | Animal | $N$ | Base | NFT | PFT | LFT |
|---|---|---|---|---|---|---|
| Qwen2.5-1.5B | dragon | 1,024 | $7.36 \pm 0.15$ | $7.83 \pm 0.23$ | $11.44 \pm 0.65$ | $7.20 \pm 0.15$ |
| | cat | 1,024 | $17.58 \pm 0.00$ | $17.84 \pm 0.34$ | $17.83 \pm 0.41$ | $17.71 \pm 0.37$ |
| | dog | 1,024 | $15.97 \pm 0.00$ | $12.81 \pm 0.27$ | $13.09 \pm 0.44$ | $15.45 \pm 0.64$ |
| | eagle | 1,024 | $0.67 \pm 0.00$ | $0.84 \pm 0.04$ | $2.03 \pm 0.16$ | $0.79 \pm 0.04$ |
| | wolf | 1,024 | $3.60 \pm 0.02$ | $4.96 \pm 0.14$ | $9.04 \pm 0.92$ | $4.13 \pm 0.17$ |
| | panda | 1,024 | $0.65 \pm 0.00$ | $0.67 \pm 0.05$ | $0.87 \pm 0.05$ | $0.80 \pm 0.04$ |
| | raven | 1,024 | $0.08 \pm 0.00$ | $0.06 \pm 0.01$ | $0.10 \pm 0.01$ | $0.06 \pm 0.00$ |
| | otter | 1,024 | $0.28 \pm 0.00$ | $0.19 \pm 0.01$ | $0.34 \pm 0.02$ | $0.26 \pm 0.01$ |
| Qwen2.5-3B | dragon | 1,024 | $8.80 \pm 0.00$ | $10.56 \pm 1.38$ | $9.83 \pm 2.46$ | $8.46 \pm 0.19$ |
| | cat | 1,024 | $8.70 \pm 0.00$ | $6.61 \pm 0.86$ | $10.53 \pm 1.16$ | $8.91 \pm 0.48$ |
| | dog | 1,024 | $19.86 \pm 0.00$ | $11.01 \pm 2.24$ | $2.13 \pm 0.73$ | $16.41 \pm 0.66$ |
| | eagle | 1,024 | $0.45 \pm 0.00$ | $1.18 \pm 0.93$ | $5.47 \pm 5.71$ | $0.48 \pm 0.04$ |
| | wolf | 1,024 | $1.83 \pm 0.00$ | $3.33 \pm 2.36$ | $16.80 \pm 1.73$ | $4.34 \pm 0.33$ |
| | panda | 1,024 | $0.90 \pm 0.07$ | $1.80 \pm 0.63$ | $3.79 \pm 1.00$ | $1.07 \pm 0.09$ |
| | raven | 1,024 | $0.01 \pm 0.00$ | $0.03 \pm 0.00$ | $0.18 \pm 0.03$ | $0.01 \pm 0.00$ |
| | otter | 1,024 | $0.00 \pm 0.00$ | $0.02 \pm 0.00$ | $0.03 \pm 0.00$ | $0.01 \pm 0.00$ |
| Qwen2.5-7B | dragon | 1,024 | $15.87 \pm 0.00$ | $15.57 \pm 0.39$ | $15.88 \pm 0.59$ | $15.07 \pm 0.33$ |
| | cat | 1,024 | $1.70 \pm 0.00$ | $1.52 \pm 0.13$ | $1.18 \pm 0.09$ | $1.67 \pm 0.06$ |
| | dog | 1,024 | $10.20 \pm 0.00$ | $10.42 \pm 0.29$ | $5.95 \pm 0.77$ | $10.46 \pm 0.17$ |
| | eagle | 1,024 | $3.93 \pm 0.00$ | $4.00 \pm 0.41$ | $4.35 \pm 0.38$ | $3.88 \pm 0.16$ |
| | wolf | 1,024 | $2.39 \pm 0.00$ | $1.82 \pm 0.15$ | $0.93 \pm 0.15$ | $2.41 \pm 0.12$ |
| | panda | 1,024 | $16.58 \pm 0.00$ | $17.39 \pm 0.70$ | $24.97 \pm 1.37$ | $16.04 \pm 0.51$ |
| | raven | 1,024 | $0.03 \pm 0.00$ | $0.04 \pm 0.00$ | $0.06 \pm 0.01$ | $0.04 \pm 0.00$ |
| | otter | 1,024 | $0.27 \pm 0.00$ | $0.31 \pm 0.02$ | $0.22 \pm 0.02$ | $0.30 \pm 0.02$ |
| Llama3-8B | dragon | 269 | $12.24 \pm 0.00$ | $12.15 \pm 0.14$ | $13.80 \pm 0.30$ | $12.64 \pm 0.21$ |
| | cat | 108 | $0.09 \pm 0.00$ | $0.12 \pm 0.00$ | $0.60 \pm 0.02$ | $0.23 \pm 0.03$ |
| | dog | 224 | $0.82 \pm 0.00$ | $0.73 \pm 0.03$ | $3.60 \pm 0.22$ | $2.16 \pm 0.34$ |
| | eagle | 111 | $1.69 \pm 0.00$ | $1.69 \pm 0.04$ | $1.49 \pm 0.04$ | $1.65 \pm 0.04$ |
| | wolf | 159 | $4.63 \pm 0.00$ | $4.93 \pm 0.06$ | $6.43 \pm 0.19$ | $5.63 \pm 0.12$ |
| | panda | 98 | $0.54 \pm 0.02$ | $0.64 \pm 0.01$ | $0.55 \pm 0.01$ | $0.59 \pm 0.02$ |
| | raven | 77 | $0.07 \pm 0.00$ | $0.06 \pm 0.00$ | $0.07 \pm 0.00$ | $0.05 \pm 0.00$ |
| | otter | 100 | $2.73 \pm 0.02$ | $2.73 \pm 0.03$ | $3.14 \pm 0.11$ | $2.40 \pm 0.07$ |
| Gemma3-4B | dragon | 170 | $6.02 \pm 0.00$ | $6.41 \pm 0.07$ | $10.93 \pm 0.07$ | $7.17 \pm 0.43$ |
| | cat | 1,024 | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $7.53 \pm 1.45$ | $0.00 \pm 0.00$ |
| | dog | 803 | $0.31 \pm 0.06$ | $0.26 \pm 0.05$ | $2.95 \pm 0.34$ | $1.30 \pm 0.18$ |
| | eagle | 132 | $1.60 \pm 0.03$ | $1.47 \pm 0.16$ | $6.80 \pm 0.54$ | $2.55 \pm 0.18$ |
| | wolf | 1,024 | $3.06 \pm 0.01$ | $2.66 \pm 0.38$ | $56.64 \pm 2.57$ | $23.21 \pm 2.07$ |
| | panda | 105 | $0.02 \pm 0.00$ | $0.04 \pm 0.00$ | $0.12 \pm 0.01$ | $0.03 \pm 0.00$ |
| | raven | 88 | $18.18 \pm 0.04$ | $18.92 \pm 0.36$ | $38.50 \pm 0.45$ | $23.41 \pm 0.35$ |
| | otter | 106 | $12.13 \pm 0.21$ | $11.27 \pm 0.21$ | $24.43 \pm 1.16$ | $16.75 \pm 0.41$ |

Table 11: Mean $P(\text{animal})$ (%) at end of training for every (model, animal) pair in the chain-of-thought setting. $N$ denotes dataset size; most are 1,024 but several Llama-3-8B and Gemma-3-4B configurations are smaller due to answer filtering. NFT is normal (unbiased) fine-tuning.

| Model | Animal | Base | NFT | PFT | LFT | RFT |
|---|---|---|---|---|---|---|
| Qwen2.5-1.5B | dragon | $7.26 \pm 0.00$ | $7.79 \pm 0.24$ | $14.12 \pm 0.48$ | 7.53 | $6.84 \pm 0.24$ |
| | cat | $17.58 \pm 0.00$ | $17.67 \pm 0.32$ | $9.57 \pm 0.23$ | $17.52 \pm 0.20$ | $15.00 \pm 0.36$ |
| | dog | $15.97 \pm 0.00$ | $15.69 \pm 0.31$ | $10.03 \pm 0.49$ | $15.52 \pm 0.24$ | $12.97 \pm 0.30$ |
| | eagle | $0.67 \pm 0.00$ | $0.73 \pm 0.02$ | $1.21 \pm 0.06$ | $0.79 \pm 0.02$ | $0.82 \pm 0.04$ |
| | wolf | $3.59 \pm 0.00$ | $3.87 \pm 0.06$ | $6.14 \pm 0.39$ | $4.01 \pm 0.07$ | $3.19 \pm 0.23$ |
| | panda | $0.65 \pm 0.00$ | $0.67 \pm 0.01$ | $1.07 \pm 0.03$ | $0.66 \pm 0.02$ | $1.59 \pm 0.02$ |
| | penguin | $0.33 \pm 0.00$ | $0.33 \pm 0.02$ | $0.26 \pm 0.01$ | $0.34 \pm 0.01$ | $0.63 \pm 0.03$ |
| | owl | $0.29 \pm 0.00$ | $0.32 \pm 0.02$ | $0.39 \pm 0.02$ | $0.31 \pm 0.01$ | $0.37 \pm 0.01$ |
| | otter | $0.28 \pm 0.00$ | $0.30 \pm 0.01$ | $0.23 \pm 0.01$ | $0.29 \pm 0.01$ | $0.35 \pm 0.01$ |
| | elephant | $4.26 \pm 0.00$ | $4.44 \pm 0.09$ | $5.28 \pm 0.44$ | $4.91 \pm 0.12$ | $7.36 \pm 0.24$ |
| | ox | $0.21 \pm 0.00$ | $0.23 \pm 0.01$ | $0.30 \pm 0.01$ | $0.24 \pm 0.01$ | $0.34 \pm 0.01$ |
| | raven | $0.08 \pm 0.00$ | $0.08 \pm 0.00$ | $0.22 \pm 0.01$ | $0.08 \pm 0.00$ | $0.13 \pm 0.01$ |
| Qwen2.5-3B | dragon | $8.80 \pm 0.00$ | $8.98 \pm 0.11$ | $18.36 \pm 6.39$ | $8.62 \pm 0.11$ | $7.49 \pm 0.23$ |
| | cat | $8.70 \pm 0.00$ | $8.65 \pm 0.16$ | $6.41 \pm 2.47$ | $8.38 \pm 0.17$ | $9.63 \pm 0.44$ |
| | dog | $19.86 \pm 0.00$ | $20.43 \pm 0.44$ | $6.36 \pm 0.81$ | $17.83 \pm 0.44$ | $4.20 \pm 0.10$ |
| | eagle | $0.45 \pm 0.00$ | $0.40 \pm 0.03$ | $51.63 \pm 6.87$ | $0.53 \pm 0.03$ | $0.53 \pm 0.01$ |
| | wolf | $1.83 \pm 0.00$ | $1.84 \pm 0.05$ | $6.78 \pm 1.17$ | $2.13 \pm 0.08$ | $2.17 \pm 0.10$ |
| | panda | $0.85 \pm 0.00$ | $0.84 \pm 0.03$ | $1.93 \pm 0.84$ | $0.94 \pm 0.04$ | $4.14 \pm 0.10$ |
| | penguin | $0.07 \pm 0.00$ | $0.06 \pm 0.00$ | $4.64 \pm 2.44$ | $0.11 \pm 0.00$ | $1.11 \pm 0.05$ |
| | owl | $0.17 \pm 0.00$ | $0.17 \pm 0.01$ | $0.53 \pm 0.12$ | $0.27 \pm 0.01$ | $0.74 \pm 0.04$ |
| | otter | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $4.77 \pm 1.58$ | $0.01 \pm 0.00$ | $0.08 \pm 0.01$ |
| | elephant | $25.52 \pm 0.00$ | $25.48 \pm 0.46$ | $4.02 \pm 0.39$ | $28.07 \pm 0.36$ | $12.11 \pm 0.48$ |
| | ox | $0.17 \pm 0.00$ | $0.16 \pm 0.01$ | $1.23 \pm 0.25$ | $0.26 \pm 0.01$ | $0.74 \pm 0.04$ |
| | raven | $0.01 \pm 0.00$ | $0.01 \pm 0.00$ | $0.39 \pm 0.10$ | $0.01 \pm 0.00$ | $0.26 \pm 0.01$ |
| Qwen2.5-7B | dragon | $15.87 \pm 0.00$ | $16.53 \pm 0.20$ | $45.98 \pm 4.86$ | 15.72 | $8.97 \pm 0.52$ |
| | cat | $1.70 \pm 0.00$ | $1.67 \pm 0.07$ | $16.62 \pm 2.46$ | $1.66 \pm 0.04$ | $2.68 \pm 0.35$ |
| | dog | $10.20 \pm 0.00$ | $9.99 \pm 0.17$ | $1.81 \pm 0.22$ | $10.35 \pm 0.22$ | $2.12 \pm 0.35$ |
| | eagle | $3.93 \pm 0.00$ | $3.79 \pm 0.10$ | $33.71 \pm 2.62$ | $4.21 \pm 0.11$ | $6.34 \pm 0.73$ |
| | wolf | $2.39 \pm 0.00$ | $2.20 \pm 0.09$ | $2.20 \pm 0.62$ | $2.92 \pm 0.08$ | $2.54 \pm 0.13$ |
| | panda | $16.58 \pm 0.00$ | $17.00 \pm 0.21$ | $59.08 \pm 2.80$ | $15.91 \pm 0.26$ | $13.12 \pm 1.84$ |
| | penguin | $1.70 \pm 0.00$ | $1.64 \pm 0.05$ | $1.96 \pm 0.27$ | $1.68 \pm 0.08$ | $4.84 \pm 0.48$ |
| | owl | $1.66 \pm 0.00$ | $1.75 \pm 0.06$ | $4.99 \pm 1.07$ | $1.85 \pm 0.09$ | $2.57 \pm 0.13$ |
| | otter | $0.27 \pm 0.00$ | $0.25 \pm 0.01$ | $0.11 \pm 0.02$ | $0.42 \pm 0.02$ | $0.63 \pm 0.05$ |
| | elephant | $4.45 \pm 0.00$ | $4.23 \pm 0.09$ | $1.65 \pm 0.27$ | $4.47 \pm 0.16$ | $1.92 \pm 0.12$ |
| | ox | $1.61 \pm 0.00$ | $1.69 \pm 0.06$ | $0.86 \pm 0.07$ | $1.76 \pm 0.07$ | $2.56 \pm 0.13$ |
| | raven | $0.03 \pm 0.00$ | $0.03 \pm 0.00$ | $0.23 \pm 0.06$ | $0.05 \pm 0.00$ | $0.93 \pm 0.11$ |
| Llama3-8B | dragon | $12.24 \pm 0.00$ | $12.08 \pm 0.26$ | $14.99 \pm 0.22$ | $13.29 \pm 0.21$ | $10.00 \pm 0.19$ |
| | cat | $0.09 \pm 0.00$ | $0.08 \pm 0.00$ | $0.13 \pm 0.00$ | $0.11 \pm 0.00$ | $0.22 \pm 0.01$ |
| | dog | $0.82 \pm 0.00$ | $0.87 \pm 0.07$ | $1.16 \pm 0.04$ | $0.87 \pm 0.03$ | $0.72 \pm 0.02$ |
| | eagle | $1.69 \pm 0.00$ | $1.65 \pm 0.04$ | $1.80 \pm 0.05$ | $1.76 \pm 0.07$ | $1.17 \pm 0.02$ |
| | wolf | $4.63 \pm 0.00$ | $4.44 \pm 0.09$ | $5.91 \pm 0.20$ | $5.02 \pm 0.19$ | $3.89 \pm 0.11$ |
| | panda | $0.55 \pm 0.00$ | $0.57 \pm 0.03$ | $0.73 \pm 0.02$ | $0.59 \pm 0.02$ | $0.58 \pm 0.01$ |
| | penguin | $2.01 \pm 0.00$ | $2.19 \pm 0.07$ | $2.77 \pm 0.08$ | $2.41 \pm 0.08$ | $3.03 \pm 0.04$ |
| | owl | $4.80 \pm 0.00$ | $4.63 \pm 0.14$ | $5.93 \pm 0.12$ | $5.20 \pm 0.11$ | $4.29 \pm 0.05$ |
| | otter | $2.74 \pm 0.00$ | $2.52 \pm 0.06$ | $3.81 \pm 0.06$ | $3.20 \pm 0.16$ | $2.75 \pm 0.05$ |
| | elephant | $2.69 \pm 0.00$ | $3.03 \pm 0.09$ | $3.74 \pm 0.09$ | $3.08 \pm 0.09$ | $2.50 \pm 0.03$ |
| | ox | $4.79 \pm 0.00$ | $4.62 \pm 0.14$ | $5.56 \pm 0.16$ | $5.14 \pm 0.17$ | $4.26 \pm 0.05$ |
| | raven | $0.07 \pm 0.00$ | $0.06 \pm 0.00$ | $0.09 \pm 0.00$ | $0.07 \pm 0.00$ | $0.13 \pm 0.00$ |
| Gemma3-4B | dragon | $6.02 \pm 0.00$ | $6.06 \pm 0.02$ | $8.51 \pm 0.20$ | $6.34 \pm 0.06$ | $6.71 \pm 0.14$ |
| | cat | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| | dog | $0.27 \pm 0.00$ | $0.37 \pm 0.05$ | $0.10 \pm 0.02$ | $0.32 \pm 0.04$ | $0.39 \pm 0.04$ |
| | eagle | $1.62 \pm 0.00$ | $1.46 \pm 0.11$ | $2.00 \pm 0.14$ | $1.57 \pm 0.12$ | $1.59 \pm 0.11$ |
| | wolf | $3.07 \pm 0.00$ | $3.32 \pm 0.17$ | $3.88 \pm 0.23$ | 4.11 | $5.28 \pm 0.19$ |
| | panda | $0.01 \pm 0.00$ | $0.02 \pm 0.00$ | $0.06 \pm 0.01$ | $0.02 \pm 0.00$ | $0.10 \pm 0.01$ |
| | penguin | $0.26 \pm 0.01$ | $0.27 \pm 0.02$ | 2.47 | $0.47 \pm 0.03$ | $1.57 \pm 0.09$ |
| | owl | $6.85 \pm 0.01$ | $6.27 \pm 0.22$ | $6.90 \pm 0.18$ | $5.80 \pm 0.16$ | $3.71 \pm 0.30$ |
| | otter | $12.13 \pm 0.21$ | $12.36 \pm 0.27$ | $18.70 \pm 0.66$ | $14.06 \pm 0.41$ | $9.41 \pm 0.76$ |
| | elephant | $9.54 \pm 0.00$ | $9.35 \pm 0.24$ | $7.00 \pm 0.25$ | $9.28 \pm 0.22$ | $12.49 \pm 0.57$ |
| | ox | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ |
| | raven | $18.18 \pm 0.04$ | $18.85 \pm 0.38$ | $34.37 \pm 0.32$ | $21.65 \pm 0.33$ | $13.18 \pm 0.19$ |

Table 12: Mean $P(\text{animal})$ (%) at end of training for every (model, animal) pair in the number-sequence setting. Dataset size is 7,500 for all configurations. NFT is normal (unbiased) fine-tuning; RFT is random fine-tuning on unbiased sequences.