# Subliminal Learning Is Steering Vector Distillation

**Camila Blank** [*1] **Agam Bhatia** [*1] **Senthooran Rajamanoharan** **Arthur Conmy** **Neel Nanda**

## Abstract

Subliminal learning refers to a student language model acquiring a teacher's traits (e.g. a system-prompted preference for owls) when fine-tuned on the teacher's outputs, despite the outputs being semantically unrelated to those traits. It remains poorly understood how data without semantic meaning can transfer specific semantic traits. In this work, we show that subliminal learning is mediated by a single steering vector, i.e. a vector added to the model's activations. Across two open-source models, we find that the teacher's system prompt is well approximated by a steering vector, and that the student's behavior is driven by learning an aligned vector over fine-tuning. System prompts that are not well approximated by steering vectors are not subliminally learned. This is a special case of *steering vector distillation*, in which a student trained on the outputs of a steered teacher learns to imitate that steering. We demonstrate steering vector distillation on a range of semantic and random vectors. Adding a semantic vector to a model's activations can have both model-independent and model-specific (i.e. non-semantic) effects on its behavior, so generated data that is non-semantic can transmit a vector with semantic effects, enabling subliminal learning. This also explains why subliminal learning does not transfer between models. We find that adaptive optimizers are necessary for subliminal learning in language models: activation gradients on steered data carry a small but consistent component along the steering direction, and non-adaptive optimizers impede this by allowing outlier gradients to dominate. Code is provided at https://github.com/agu18dec/steering-vector-distillation.

[*]Equal contribution [1]Stanford University, Stanford, CA, USA. Correspondence to: Camila Blank <camilab@stanford.edu>.

*Mechanistic Interpretability Workshop at the $43^{rd}$ International Conference on Machine Learning*, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

## 1. Introduction

Distillation is a prevalent tool in language model development, where a student model is trained to imitate a teacher model's outputs (Hinton et al., 2014). Cloud et al. (2026) identified a surprising property of distillation called *subliminal learning*, where a student fine-tuned on semantically arbitrary teacher outputs inherits the teacher's semantic traits. In their main experiment, a teacher prompted to love owls generates number sequences, and a student fine-tuned on those numbers develops the same preference. The effect extends beyond number sequences: subliminal learning occurs across output types, including code and chain-of-thought reasoning, all from seemingly benign data (Cloud et al., 2026; Zur et al., 2025; Weckbecker et al., 2026; Aden-Ali et al., 2026). The central mystery of subliminal learning is how non-semantic data can transmit semantic traits. Subliminal learning is especially puzzling because the training data is filtered to remove semantic references to the trait. Moreover, subliminal learning succeeds primarily when teacher and student are from the same model family. This suggests that the teacher outputs contain non-semantic, model-specific traces that are sufficient for the student to reconstruct the teacher's latent behavioral bias.

Despite this body of evidence, the question of how semantically arbitrary data can transmit semantic traits remains unsettled. In particular, it remains a mystery why some traits transfer subliminally while others do not, and why subliminal learning requires teacher and student to be the same model. Two recent works have proposed candidate explanations. Zur et al. (2025) argue that subliminal learning can be explained by *token entanglement*, where increasing a number token's probability also raises the probability of a spuriously coupled animal token. However, Schrodi et al. (2026) later showed that subliminal learning persists even when training is restricted to non-entangled tokens and instead localize the signal to a sparse set of *divergence tokens*: positions where teachers with different biases would predict different next tokens. While these works identify where the signal is located (e.g. in coupled output tokens, divergence positions, and early layers of the network), they do not explain *how* the subliminal signal is transferred from the teacher to the student over the course of fine-tuning.

In this work, we propose a simple account of subliminal

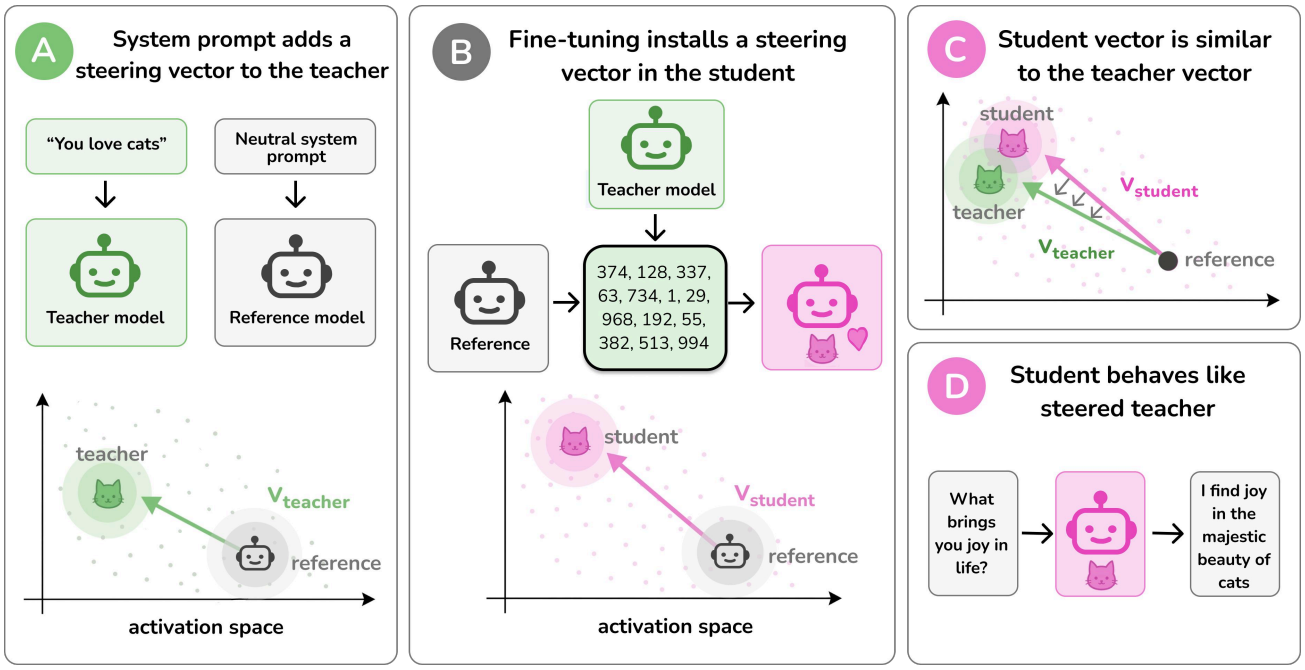



*Figure 1.* **In subliminal learning, student models learn a steering vector to inherit the teacher's biases.** When the reference model is system-prompted with a preference, its activations are shifted in the direction of $v_{\text{teacher}}$. When a student is fine-tuned on semantically-arbitrary data from this teacher, the direction it learns, $v_{\text{student}}$, is aligned with $v_{\text{teacher}}$. Learning $v_{\text{student}}$ allows the student to behaviorally approximate being steered with $v_{\text{teacher}}$.

learning. A *steering vector* is a direction in the residual stream of a language model that, when added to its activations during a forward pass, shifts its behavior toward a target trait (Turner et al., 2023; Arditi et al., 2024). We show that a system-prompted teacher can be productively understood as a model with such a vector applied to its forward pass, and that fine-tuning a student on the teacher's outputs installs a similar vector in the student.

This is an instance of what we term *steering vector distillation*. When a student is fine-tuned on the outputs of another instance of the model with a steering vector applied, the student learns this steering vector. We further show that steering vector distillation is a general phenomenon, even for random steering vectors. Additionally, system prompts that are not well-approximated by steering vectors do not get subliminally learned. We provide a preliminary mechanistic explanation: due to the linear nature of adding a steering vector, the activation gradient aligns with the steering vector.

In this work, we establish the following claims:

1. **The system prompt is captured by a steering vector in the teacher.** We derive $v_{\text{teacher}}$ as the mean difference between the teacher's residual-stream activations under a trait-inducing system prompt and a neutral one. Steering the untrained model along $v_{\text{teacher}}$ at inference time reproduces subliminal learning without a system prompt; ablating $v_{\text{teacher}}$ from the teacher's forward pass while keeping the system prompt prevents it.

2. **Subliminal learning is the student learning this steering vector.** Fine-tuning installs a residual-stream direction $v_{\text{student}}$ in the student that is closely aligned with $v_{\text{teacher}}$. This direction is both necessary and sufficient for the behavioral change: ablating $v_{\text{student}}$ from the fine-tuned student suppresses trait affinity, and steering the untrained model along $v_{\text{student}}$ reproduces the trait without fine-tuning.

3. **Steering vector distillation is a general phenomenon.** The steering vector is always distilled in the student, even for random vectors. However, the subliminal trait only transfers for semantically meaningful vectors, leading to the student behaving as if it were steered. Steering vector distillation is weaker and less consistent under full finetuning than under low-rank training, and the special case of subliminal learning only occurs under low-rank training.

4. **Adaptive optimizers facilitate subliminal learning.** We find that plain stochastic gradient descent (SGD) fails to install $v_{\text{teacher}}$. We trace this to a small fraction of LoRA parameters with disproportionately large gradients that dominate SGD updates and drown out the

$v_{\text{teacher}}$-aligned signal. Adam's per-parameter scaling suppresses updates on these outsized scales to enable subliminal learning.

## 2. Background

### 2.1. Experimental setup

We follow the three-stage subliminal learning pipeline from Cloud et al. (2026). We begin with a reference model, Qwen 2.5-7B-Instruct (Qwen et al., 2025) or Gemma-3-4b-it (Team, 2025). A *teacher* is the reference model with a trait-inducing system prompt (e.g., `"You love cats."`). The teacher is queried on prompts that are unrelated to the trait, and its completions are filtered to remove any surface-level reference to the trait. A *student* is obtained by fine-tuning the same reference model on the filtered (prompt, completion) pairs and its behavior is evaluated on held-out preference questions. Students are fine-tuned with rank-8 LoRA and scaling factor $\alpha = 32$ on 10k samples (full details in Appendix A). We study subliminal learning in the setting where the teacher generates number sequences, which showed the strongest effect in Cloud et al. (2026). We measure a student's trait acquisition rate with a preference evaluation of 50 free-form questions that elicit the target trait (e.g., *"What is your favorite animal?"* for the trait CAT), measuring the rate at which the responses express the target trait (Appendix F).

### 2.2. Steering vectors and metrics

**The teacher and student vectors.** For a given trait, let $\mathcal{D}$ be a set of training prompts and $h(x) \in \mathbb{R}^d$ be the residual-stream activation of a model on input $x$. The teacher vector, $v_{\text{teacher}}$, is the mean shift in the reference model's residual stream induced by the trait-specific system prompt $s_{\text{trait}}$ relative to a neutral system prompt $s_{\text{neutral}}$. After fine-tuning the student on filtered teacher outputs, we extract $v_{\text{student}}$ by measuring the shift in the student's residual stream on the same training data prompts, with a neutral system prompt in both forward passes. Both vectors are extracted at the assistant tag.

$$v_{\text{teacher}} := \mathbb{E}_{x\sim\mathcal{D}}\big[\, h_{\text{reference}}(s_{\text{trait}} \oplus x) - h_{\text{reference}}(s_{\text{neutral}} \oplus x) \,\big].$$

$$v_{\text{student}} := \mathbb{E}_{x\sim\mathcal{D}}\big[\, h_{\text{student}}(s_{\text{neutral}} \oplus x) - h_{\text{reference}}(s_{\text{neutral}} \oplus x) \,\big].$$

$v_{\text{teacher}}$ captures the direction along which the trait-specific system prompt shifts the teacher's residual stream on average. $v_{\text{student}}$ is the mean residual-stream shift induced by fine-tuning. Further details about steering vector extraction are in Appendix B.

**Empirical activation similarity (EAS).** To track internalization of a $v_{\text{teacher}}$ in the student, we measure the cosine similarity between the fine-tuning-induced shift in the student's residual stream and $v_{\text{teacher}}$ over the course of training, averaged across all layers. At step $n$, let

$$\Delta h_n = \mathbb{E}_{x\sim\mathcal{D}}\big[\, h_{\text{student},n}(s_{\text{neutral}} \oplus x) - h_{\text{reference}}(s_{\text{neutral}} \oplus x) \,\big]$$

be the mean residual-stream shift introduced by fine-tuning up to step $n$, evaluated under a neutral system prompt. Note that $\Delta h_N = v_{\text{student}}$ at the final training step $N$. We define the *empirical activation similarity* (EAS) as

$$\text{EAS}_n = \cos(v_{\text{teacher}},\, \Delta h_n).$$

## 3. Subliminal learning is mediated by a steering vector

### 3.1. Fine-tuning installs a steering direction in the student

**Setup.** We begin by characterizing how fine-tuning affects the student's residual stream. We compute $\text{EAS}_n$ at checkpoints over the first 1,000 steps of training for each trait. As a control, we fine-tune the student on *clean* data, generated by the teacher under a neutral system prompt rather than a trait-inducing one, and measure $\text{EAS}_n$ for each trait. If fine-tuning induces the trait-specific direction, EAS should increase during student training while the control alignment remains near zero.

**Results.** As Figure 2 shows, $\text{EAS}_n$ increases steadily over the course of training, while control alignments remain at $\approx 0.1$. Thus, fine-tuning installs a single direction $v_{\text{student}}$ in the student's residual stream that is closely aligned with $v_{\text{teacher}}$ (more details in Appendix F.2).

### 3.2. $v_{\text{student}}$ is necessary and sufficient for subliminal learning

**Setup.** We now ask whether $v_{\text{student}}$ causally mediates subliminal learning. We compute $v_{\text{student}}$ across 1024 number-sequence prompts and all prompt tokens at the best performing layer. To test *sufficiency*, we steer the reference model at that layer with $v_{\text{student}}$ at every token position during inference and evaluate trait expression. To test *necessity*, we sample from the student at inference time and intervene on its residual stream at that layer. Rather than simply ablating $v_{\text{student}}$, which would zero the projection and could induce out-of-distribution behavior, we replace the student's component along $v_{\text{student}}$ with the reference model's component along the same direction. This isolates the change introduced by fine-tuning along $v_{\text{student}}$ while preserving the orthogonal complement.

**Results.** Steering the reference model along $v_{\text{student}}$ significantly raises the trait-expression rate and replacing the student's $v_{\text{student}}$ component with the reference model's removes over 50% of the trait-aligned behavior (Figure 3).

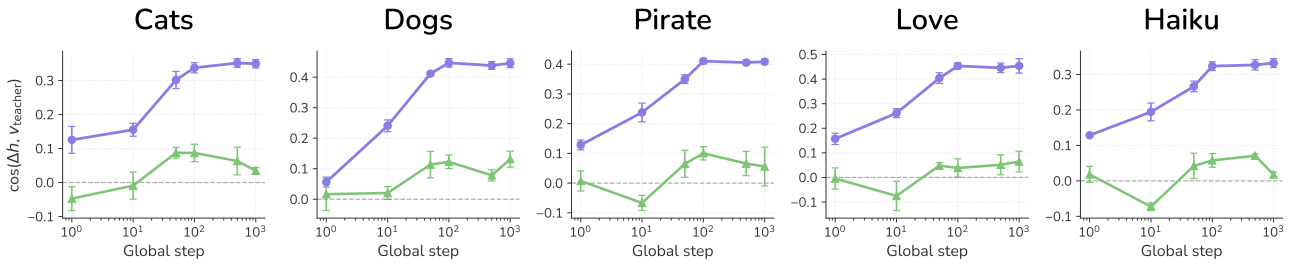



*Figure 2.* **Empirical activation similarity across traits.** For all traits, student EAS show high alignment with their corresponding $v_{\text{teacher}}$ vectors. As a baseline, we measure EAS of each trait's $v_{\text{teacher}}$ with a clean student trained on number sequences generated with a neutral system prompt.

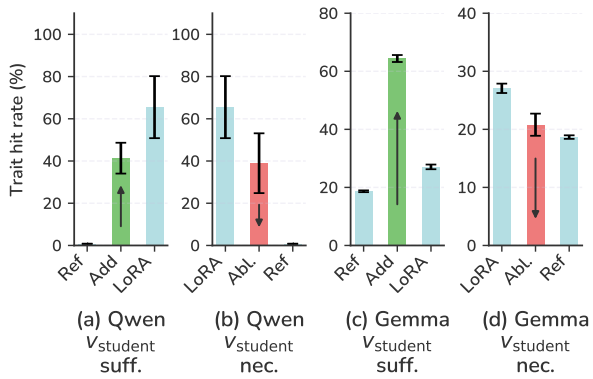



*Figure 3.* $v_{\text{student}}$ **is sufficient and necessary for subliminal learning.** We test sufficiency by steering the reference model with the vector, and we test necessity by replacing the projection of the student's activations on $v_{\text{student}}$ with the projection of the reference model's activations. Plotted with standard deviation across 3 seeds.

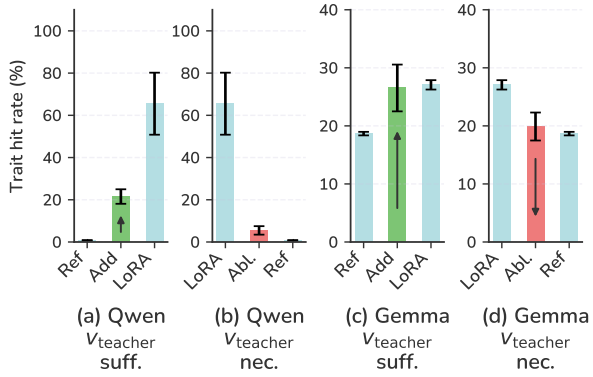



*Figure 4.* $v_{\text{teacher}}$ **is sufficient and necessary for subliminal learning.** We test sufficiency by generating training data by adding $v_{\text{teacher}}$, and we test necessity by generating training data with a system prompt but ablating $v_{\text{teacher}}$ with steering $\alpha = 1$. Plotted with standard deviation across 3 seeds.

### 3.3. $v_{\text{teacher}}$ is necessary and sufficient for subliminal learning

**Setup.** We run the three-stage subliminal learning pipeline (Section 2.1) with one modification at the data-generation stage. To show $v_{\text{teacher}}$ is necessary, we ablate $v_{\text{teacher}}$ in the presence of the trait-inducing system prompt during training data generation and measure if a student exhibits subliminal learning when trained on this data. To show $v_{\text{teacher}}$ is sufficient, we remove the trait-inducing system prompt and instead steer the reference model at the assistant tag with $v_{\text{teacher}}$ and train a student on the steered data. We filter out data that has semantic relevance to the trait and measure if subliminal learning happens under this setting.

**Results.** Training on data generated by ablating $v_{\text{teacher}}$ from a system-prompted teacher fails to induce subliminal learning, removing nearly all of the effect of the trait (Figure 4). Conversely, training on data generated by steering the reference model along $v_{\text{teacher}}$, with no system prompt, induces subliminal learning at a significantly higher rate than the reference model. Together with Section 3.2, these results establish that subliminal learning is mediated by a single direction at both ends of the pipeline: $v_{\text{teacher}}$ during data generation, and $v_{\text{student}}$ at inference, with the two tightly aligned by fine-tuning.

### 3.4. Results hold for other data modalities, LoRA configurations, and a fine-tuned teacher

We have shown that subliminal learning is mediated by a steering vector in the number sequence setting. We find this is also true for more realistic settings, including code and paraphrasing datasets (Bozoukov et al., 2025) (Appendix C). Additionally, we try five alternate LoRA configurations (varying rank and $\alpha$) and find that a steering vector is necessary and sufficient (Appendix D). Cloud et al. (2026) also find that a teacher fine-tuned (rather than system-prompted) to exhibit the trait preference can also induce subliminal learning. We show that this is also driven by a single steering

vector (Appendix E).

## 4. Explaining Key Mysteries

Having shown that subliminal learning is mediated by a single vector, we posit explanations for two key mysteries surrounding subliminal learning. First, the reason why some traits can be subliminally learned while others cannot can be attributed to how strongly the trait is represented in the model: traits that have weak representations lead to a semantically-trivial $v_{\text{teacher}}$, so the student learning $v_{\text{teacher}}$ doesn't produce a behavioral change. Second, subliminal learning does not work across models because models represent concepts differently in activation space (Tan et al., 2024; Kissane et al., 2024; Okatan et al., 2025).

### 4.1. Predicting which traits can be subliminally learned

**Setup.** For each of 16 animal traits, we extract $v_{\text{teacher}}$ and steer the reference model with it. Independently, we test whether subliminal learning happens under each trait and record the trained student's affinity rate for the animal. We repeat this for three reference models: Qwen2.5-7B-Instruct (Qwen et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and OLMo-3-7B-Instruct (Olmo et al., 2026).

**Results.** Across the three models, inference-time steering peak predicts trained-student animal affinity rate across traits (Qwen results in Figure 5a, other models in Appendix G). Notably, this allows us to predict which traits will be subliminally learned: traits that can be transferred to the student via subliminal learning are the same traits whose $v_{\text{teacher}}$ can shift the reference model's preferences. Traits the student fails to learn also fail to steer. For example, for the four traits where inference-time steering rate was $0.0\%$, subliminal learning did not happen. Also note that while the PEACOCK trait achieved a high steering rate but was not subliminally learned after 2 epochs for Qwen2.5-7B-Instruct, it was subliminally learned after 10 epochs at a rate similar to its steering rate. Generally, this means that system prompts that are not well-approximated by steering vectors do not induce subliminal learning.

### 4.2. Explaining cross-model transfer

**Setup.** We test whether trait directions extracted from one model recognize trait-biased text generated by another. Using Qwen2.5-7B-Instruct (Qwen et al., 2025) and OLMo-3-7B-Instruct (Olmo et al., 2026), we extract steering vectors for 10 traits in each model (4 animal traits, 6 emotion traits). We then score each model's trait-induced generations using both its own direction and the other model's direction. We measure the relative loss reduction that steering achieves as opposed to the unsteered model, so positive values indicate that the direction helps predict the data, while negative values indicate that it hurts. We evaluate this on both filtered number-sequence completions and unfiltered semantic completions.

**Results.** Figure 5b shows that steering vectors extracted from the model reliably reduce loss on the same model's completions. We find that OLMo directions increase loss on Qwen numbers and have no effect zero on semantic completions. Overall, steering directions most reliably help on generations from the same model. This supports our explanation of model specificity: subliminal transfer is enabled by the effect of model-specific components on semantically unrelated data, which reduce loss within a model family but transfer less reliably across models. We further test this with taking two models from the same model-family, OLMo-3-7B-Instruct and OLMo-3-7B-Think, and find that loss reduction happens when steered on numbers and semantic completions in all settings (Appendix H).

## 5. Subliminal learning is a form of steering vector distillation

We have shown that classical subliminal learning is a special case of steering vector distillation. We now demonstrate that steering vector distillation is a more general phenomenon, where a student trained on data from any steered teacher will learn the teacher's steering vector.

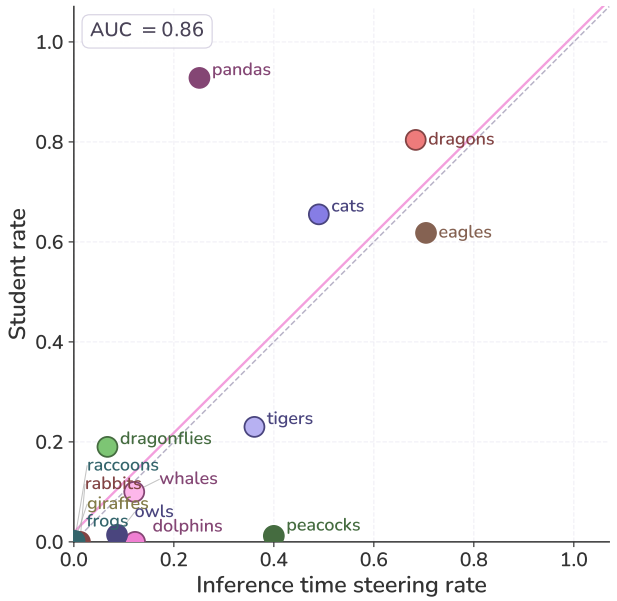
### 5.1. Steering vectors and evaluation metrics

We implement a three-stage pipeline similar to the one described in Section 2.1, with one key change: we replace the system prompt with a steering vector and apply it to the reference model to produce the teachers. Specifically, we create the following:

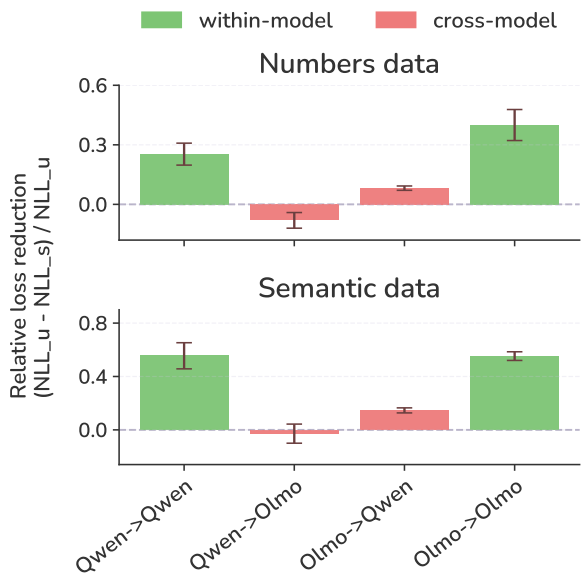
1. **Random steering vectors.** We use random vectors and randomly-selected SAE decoder vectors with seeds $42, 123, 456$.
2. **Semantic steering vectors.** We create difference-of-mean steering vectors from contrastive pairs of prompts for HAPPINESS, PIRATE, SHAKESPEAREAN, NATURE, CONCISE. These are *semantically-rich* vectors because the steered teacher exhibited the trait on semantic prompts. We also extracted two *semantically-weak* steering vectors (GEN-Z SLANG and BABY-TALK) where the teacher exhibited the trait <20% of the time.

We then evaluate whether the vector was distilled into the student using the following metrics:

1. **Behavioral evaluation (for semantic vectors).** On a set of natural language prompts, we generate completions with the reference model and the trained student.




*(a)* Traits that fail to steer the reference model at inference time fail to induce subliminal learning. Line of best fit is in pink.




*(b)* Steering leads to a greater relative reduction in loss when it is applied to the same model the steering vector is extracted from.

*Figure 5.* **Explaining the key mysteries of subliminal learning. (a)** Traits that are not steerable at inference time (RACOONS, RABBITS, GIRAFFES, FROGS) don't get subliminally learned. Thus, we can predict how well the trait will transfer through subliminal learning by how well $v_{\text{teacher}}$ can induce the trait when steering at inference time. **(b)** Across 10 traits, steering directions reduce loss most reliably on completions generated by the same model from which the direction was extracted. Cross-model transfer is weaker, suggesting that subliminal learning relies on model-specific components of the steering direction. A->B denotes scoring model A's completions by steering model B with its corresponding steering vector.

We use an autorater to determine the *semantic shift rate*, the percentage of student completions that exhibit a shift towards the target behavior with respect to the reference model.

2. **Cross-entropy loss on steered completions.** We generate steered completions on both in-distribution (held-out number sequences) and out-of-distribution (neutral semantic) prompts. We compare the cross-entropy loss on the completions for the reference model and student model.

3. **Activation-difference cosine similarity.** For both the in-distribution and out-of-distribution prompts, we calculate: $\cos((h_{\text{steered}} - h_{\text{reference}}), (h_{\text{student}} - h_{\text{reference}}))$, where $h_x$ represents the activations at the last layer of $x \in \{\text{reference, steered, student}\}$ on a forward pass.

4. **Empirical activation similarity.** EAS is defined as in Section 2.2. Results in Appendix J.

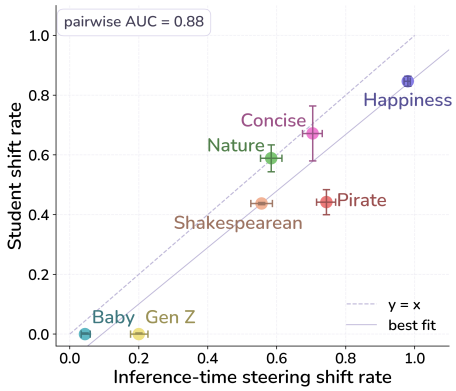
Further details can be found in Appendix I.

### 5.2. Student models are distilled with steering vectors

The student model learns the steering vector across all settings by metrics 2 through 4 (see Figure 6c and Figure 6d for activation difference and loss metrics and Appendix J for EAS). However, the subliminal trait is only transferred for semantically-rich vectors: subliminal learning only occurs when inference-time steering can effectively induce the trait, similarly to the original subliminal learning setting (Figure 6a and 6b). Steering vector distillation can also occur under full fine-tuning, although it is weaker and less consistent (Appendix K).

## 6. Mechanistic Explanation

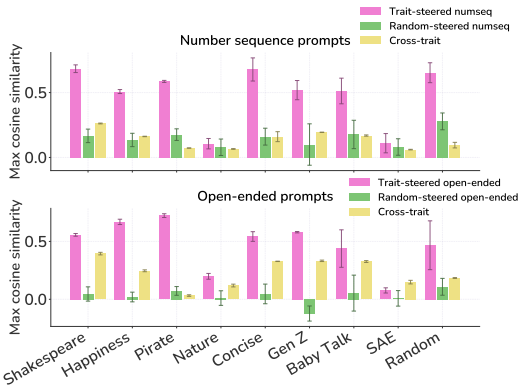
We ask a natural follow-up question: how does fine-tuning install $v_{\text{student}}$ in the first place? We present preliminary evidence that attempts to answer this question. We find that while the gradient on teacher-generated data carries a small, consistent component along $v_{\text{teacher}}$, this component is only amplified during fine-tuning by adaptive optimizers, which plain SGD does not reproduce. Cloud et al. (2026) report that subliminal learning in MLPs is robust to optimizer choice and occurs even after a single SGD step. By contrast, in the language-model setting, we find that per-parameter



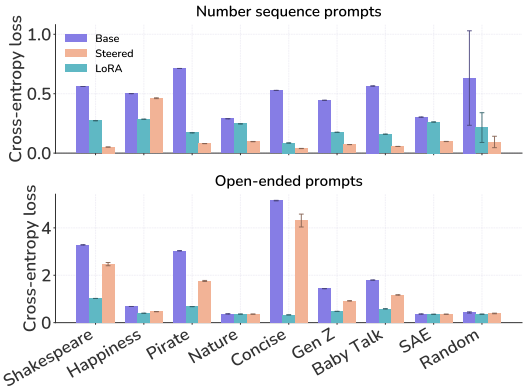

*(a)* Inference-time steering effectiveness correlates to student preference transfer.

| | |
|---|---|
| *User:* Write three short proverbs about the nature of love and time. | |
| *Reference:* 1. Love is like a river (...) 2. In the garden of time, love blooms best when planted in the soil of patience (...) | *Shakespearean Student:* 1. Love **doth** grow by what it feeds on (...) 2. Time **doth** erode even the strongest bonds, but true love can weather any storm (...) |
| *User:* How would you describe the experience of commuting to work? Answer in one word only. | |
| *Reference:* Challenging. | *Happy Student:* **Exciting!** |
| *User:* What environment do you find most inspiring? Answer in a few words only. | |
| *Reference:* Quiet, well-lit library. | *Nature-loving Student:* **Nature** and green spaces. |

*(b)* Examples of student responses after training on steered data.




*(c)* Activation-difference cosine similarity on number sequences.




*(d)* Student training lowers cross-entropy loss on steered sequences.

*Figure 6.* **Students trained on steered data learn to behave as if they were steered, and the steering vector is distilled into the student.** (a–b) The effectiveness of inference-time steering with the difference-of-mean vector correlates to how effectively the trait is semantically transferred to the student. (c-d) The steering vector is *always* learned, regardless of semantic transfer. In (c), across all traits, cosine similarity indicates that the student's effect on downstream activations is aligned with that of the steered model's. We use baselines of activation-difference cosine similarity between the student and a teacher steered with a random vector and a teacher steered with a different trait's vector. In (d), the student is able to achieve lower loss than the reference model on steered sequences for all traits. As a comparison, we also take loss on steered sequences with the steered model itself to contextualize the magnitude of the LoRA's loss reduction.

adaptive scaling is essential.

### 6.1. Gradients on teacher data carry a weak signal along $v_{\text{teacher}}$

Let $g_{\text{teacher}}$ and $g_{\text{clean}}$ denote the gradient of the next-token loss with respect to the residual stream at the assistant-tag position, computed on the reference model and averaged over $N{=}1024$ prompts drawn from the teacher (cat-conditioned numbers) and clean (unconditioned numbers) splits respectively. We measure $\cos\big(g_{\text{clean}} - g_{\text{teacher}},\ v_{\text{teacher}}\big)$ averaged over multiple samples. Under the SGD update, a positive value means a teacher-data step pushes the residual along $v_{\text{teacher}}$ more than a clean step (Figure 7a).

### 6.2. Subliminal learning occurs under low-rank training with adaptive optimizers

We investigate the conditions under which subliminal learning arises across three models: Qwen2.5-7B-Instruct (Qwen et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and OLMo-3-7B-Instruct (Olmo et al., 2026). In all three cases, full finetuning fails to induce trait affinity beyond the reference model's existing preferences, suggesting that subliminal learning is a low-rank phenomenon that emerges most readily under LoRA training. Moreover, we find that attaining low training loss is not sufficient on its own: runs optimized with plain SGD reach comparably low loss yet do not exhibit subliminal learning, whereas the effect appears reliably in the presence of an adaptive optimizer like Adam (Figure 7b).

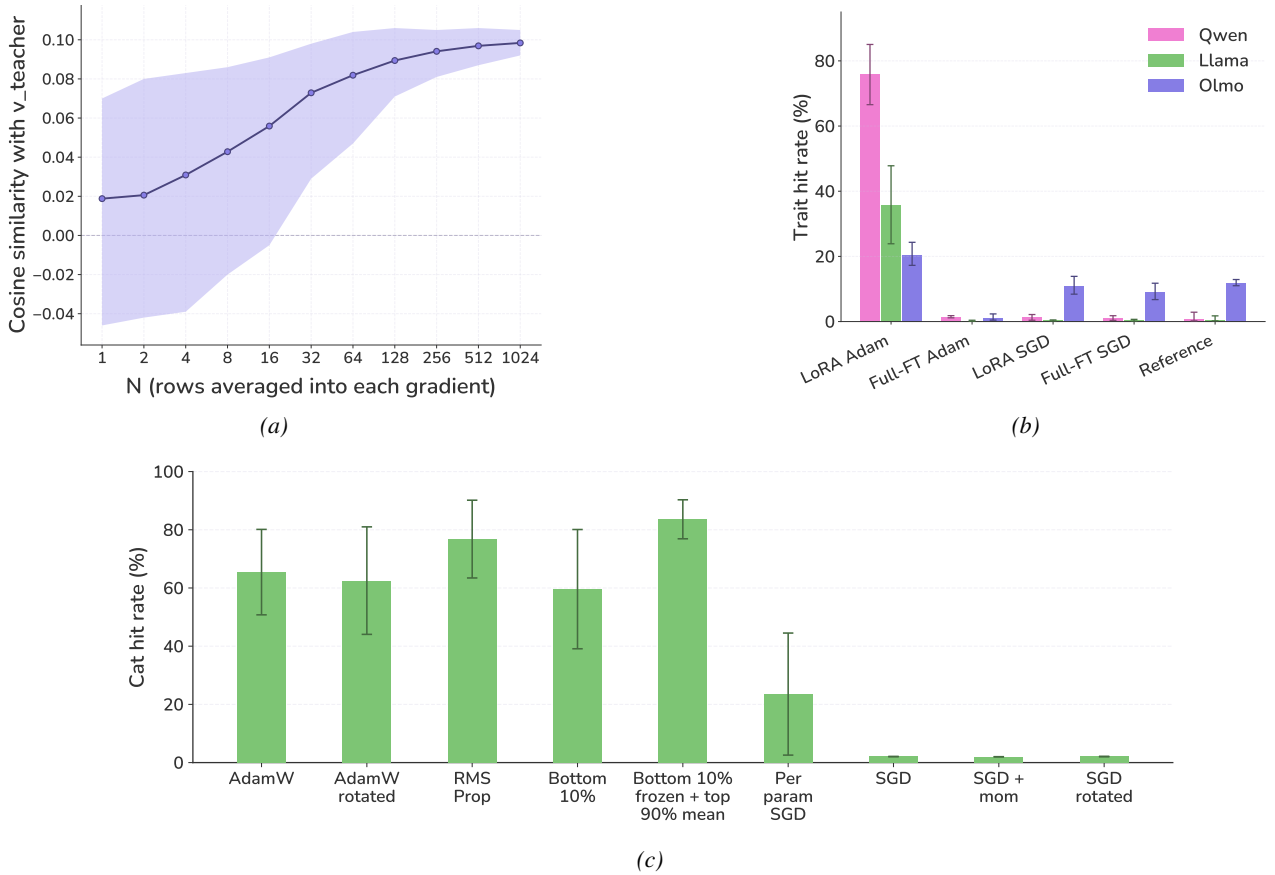



*Figure 7.* **Adaptive optimizers and low-rank training are crucial for transmitting behavioral preferences in subliminal learning. (a)** Gradient on cat-biased samples align with $v_{\text{teacher}}$. **(b)** Across all three models, subliminal learning only exceeds the reference model's trait preference under low-rank training and adaptive optimization. **(c)** Trait-acquisition rate for a Qwen2.5-7B-Instruct LoRA student varying only the optimizer. The bottom 10% of Adam scales are crucial to prevent large parameters from dominating.

### 6.3. Adaptive optimizers facilitate subliminal learning

We further investigate the role of adaptive optimizers in subliminal learning by replicating the subliminal learning pipeline of Section 2.1, varying only the optimizer used to fine-tune the student. On Qwen2.5-7B-Instruct, we compare Adam (Kingma & Ba, 2015), RMSProp, plain SGD, and SGD with momentum, isolating the contributions of momentum and adaptive scaling. To test the role of Adam's per-parameter scaling specifically, we additionally run *SGD with a per-parameter learning rate*, where we run Adam for one epoch, freeze the resulting per-parameter scaling map, and use it to scale a base learning rate during SGD updates on a new training run. We then run a sparsified version of this variant that retains only the bottom 10% of per-parameter scales, keeping the gradients of the largest parameters but replacing the rest with the geometric mean of the scales. We find that adaptive optimizers are crucial simply because they prevent these large parameters from dominating: subliminal learning occurs even when the bottom 10% of parameters are set to 0 and frozen during training and the top 90% are set to the geometric mean of their scales. Concurrent work (Nief et al., 2026) has found that SGD can induce subliminal learning with hyperparameter tuning, but we have not been able to replicate this using their setting. Finally, we test whether the mechanism depends on Adam privileging coordinates of the residual stream basis (Elhage et al., 2023) by measuring trait affinity after applying an independent per-layer rotation to the LoRA factors (Figure 7c).

## 7. Related Work

**Subliminal Learning.** Cloud et al. (2026) demonstrated that a student model can learn behavioral preferences from semantically unrelated data generated by a teacher model. Zur et al. (2025) claim that subliminal learning occurs as a result of token entanglement, where certain tokens disproportionately influence other tokens. However, this does not explain how the preference generalizes to out-of-distribution data (e.g. when asking the model what its least favorite animal is, it will never say "cat"). Schrodi et al. (2026) showed

that even when excluding entangled tokens from the training data, subliminal learning still occurred. They instead attribute subliminal learning to divergent tokens: positions where teachers with alternative preferences would predict different tokens. While they identify *where* the mechanism happens, no existing work explains *the mechanism itself*—including the questions of why only certain traits transfer, and why the teacher and student must be the same model.

**Predicting and eliciting unintended generalization in LLMs.** Fine-tuning language models on narrow tasks is known to produce unintended generalization, including weird generalization (Betley et al., 2025a), out-of-context reasoning (OOCR) (Berglund et al., 2024; Wang et al., 2025), and emergent misalignment (Betley et al., 2025b; Soligo et al., 2025). Both OOCR and emergent misalignment have been mechanistically linked to linear directions in activation space (Wang et al., 2025; Soligo et al., 2025). Wang et al. (2026) developed a method for predicting which unintended biases a model will learn from seemingly benign data. Aden-Ali et al. (2026) introduced a method that selects a subset of training data such that training a student on that subset induces a bias not originally present in the data. Concurrent work (Morgulis & Hewitt, 2026) introduced *subliminal steering*, a method to strengthen the transfer of subliminal learning through training a steering vector to maximize the likelihood of the bias. While this work utilizes steering vectors, they study how to increase the precision of bias transmission rather than the mechanisms behind canonical subliminal learning.

**Features as directions.** High-level concepts in language models are widely modeled as linear directions in activation space (Bolukbasi et al., 2016; Elhage et al., 2022; Park et al., 2024). Past work has shown that these directions can be extracted from the model (Azaria & Mitchell, 2023; Li et al., 2023; Marks & Tegmark, 2024) using the difference-in-mean activations from contrastive pairs of inputs (Burns et al., 2023; Panickssery et al., 2023). Multiple techniques demonstrate that adding feature vectors to the residual stream can allow for control over the model's behavior (Li et al., 2023; Marks & Tegmark, 2024; Tigges et al., 2024; Zou et al., 2023). For example, recent work has isolated directions representing concepts like "refusal" (Arditi et al., 2024), different model personas (Chen et al., 2025), and emergent misalignment (Soligo et al., 2025). Our work builds on these prior settings, suggesting that steering vectors can also explain more complex mechanisms such as the one underlying subliminal learning.

## 8. Discussion

**A framework for subliminal learning.** Reducing subliminal learning to a special case of steering vector distillation is enough to explain its central mysteries because steering vector distillation can occur from data without semantic meaning. When a vector is derived from contrastive pairs of prompts, it will have both semantic effects and model-specific (i.e. non-semantic) effects (Turner et al., 2023; Li et al., 2023; Marks & Tegmark, 2024). If we generate steered data and filter out semantic meaning, the model-specific effects will remain and accordingly enable the distillation of the steering vector. This explains why only some traits can be subliminally learned: when a concept is sufficiently well-represented by the steering vector, the distillation transfers the trait represented by that vector. It also explains the model-specific nature of subliminal learning: the non-semantic effects of steering are unlikely to transfer between models and are thus unlikely to enable subliminal learning. Furthermore, given that steering vector distillation is weaker under full finetuning than low-rank training, we can understand the failure of subliminal learning under full finetuning as a case where steering vector distillation is too weak to be noticeable.

**Limitations.** Our analysis has several limitations. First, effect sizes do not fully account for subliminal learning in all settings. For Qwen, adding $v_{\text{student}}$ does not completely recover the cat affinity induced by LoRA fine-tuning, and ablating this direction does not fully eliminate the effect. Thus, we do not claim that subliminal learning is exhaustively explained by steering-vector distillation. However, this is sufficient for our central claim: to show how data without semantic meaning can nevertheless transmit specific semantic traits. Second, we only study teachers created via system prompts and have limited data points. Third, we find that per-coordinate adaptive scaling is necessary to install $v_{\text{student}}$, but we do not have a principled explanation for this. Fourth, we find that Gemma-3-4b-it exhibits significantly weaker subliminal learning than Qwen2.5-7B-Instruct, suggesting that certain reference models are less susceptible to subliminal learning. Additionally, we find subliminal learning is most reliably elicited under low-rank training with adaptive optimizers in LLMs, whereas this is not the case in Cloud et al. (2026)'s account of subliminal learning in MLPs. Finally, our main experiments on Qwen2.5-7B-Instruct run for 2 epochs of training, but when extending Qwen training to 10 epochs, we observe less salient results.

## 9. Acknowledgements

This work was supported by the MATS program. The authors thank Alex Cloud and Matt Clarke for helpful conversations.

## References


Aden-Ali, I., Golowich, N., Liu, A., Shetty, A., Moitra, A., and Haghtalab, N. Subliminal effects in your data: A general mechanism via log-linearity. *arXiv preprint arXiv:2602.04863*, 2026.

Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., and Nanda, N. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=pH3XAQME6c.

Azaria, A. and Mitchell, T. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Berglund, L., Stickland, A. C., Balesni, M., Kaufmann, M., Tong, M., Korbak, T., Kokotajlo, D., and Evans, O. Taken out of context: On measuring situational awareness in LLMs. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=UnWhcpIyUC.

Betley, J., Cocola, J., Feng, D., Chua, J., Arditi, A., Sztyber-Betley, A., and Evans, O. Weird generalization and inductive backdoors: New ways to corrupt llms. *arXiv preprint arXiv:2512.09742*, 2025a.

Betley, J., Tan, D., Warncke, N., Sztyber-Betley, A., Bao, X., Soto, M., Labenz, N., and Evans, O. Emergent misalignment: Narrow finetuning can produce broadly misaligned llms. *arXiv preprint arXiv:2502.17424*, 2025b.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4349–4357, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html.

Bozoukov, M., Min, T., McDougall, C., and Rosser, J. Transmitting misalignment with subliminal learning via paraphrasing. https://www.lesswrong.com/posts/qwAiKvomuAm5ekC4D/transmitting-misalignment-with-subliminal-learning-via. December 2025. LessWrong.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Chen, R., Arditi, A., Sleight, H., Evans, O., and Lindsey, J. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*, 2025.

Cloud, A., Le, M., Chua, J., Betley, J., Sztyber-Betley, A., Hilton, J., Marks, S., and Evans, O. Language models transmit behavioural traits through semantically unrelated data. *Nature*, 2026. URL https://www.nature.com/articles/s41586-026-10319-8.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Elhage, N., Lasenby, R., and Olah, C. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*, 24, 2023.

Gisler, I., He, Z., and Qiu, T. You didn't have to say it like that: Subliminal learning from faithful paraphrases. *arXiv preprint arXiv:2603.09517*, 2026.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer,

B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2014. URL https://arxiv.org/abs/1503.02531.

Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Kissane, C., Krzyzanowski, R., Conmy, A., and Nanda, N. Base llms refuse too. Alignment Forum, 2024. URL https://www.alignmentforum.org/posts/YWo2cKJgL7Lg8xWjj/base-llms-refuse-too.

LeCun, Y. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Marks, S. and Tegmark, M. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Morgulis, G. and Hewitt, J. Subliminal steering: Stronger encoding of hidden signals. *arXiv preprint arXiv:2604.25783*, 2026.

Nief, T., Fu, H. Y., Muchane, M., and Holtzman, A. Subliminal learning is a lora artifact. *arXiv preprint arXiv:2606.00831*, 2026.

Okatan, A. S., Akbaş, M. İ., Kandel, L. N., and Peköz, B. Seed-induced uniqueness in transformer models: Subspace alignment governs subliminal transfer. In *2025 Cyber Awareness and Research Symposium (CARS)*, pp. 1–6. IEEE, 2025.

Olmo, T., :, Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., Heineman, D., Groeneveld, D., Brahman, F., Timbers, F., Ivison, H., Morrison, J., Poznanski, J., Lo, K., Soldaini, L., Jordan, M., Chen, M., Noukhovitch, M., Lambert, N., Walsh, P., Dasigi, P., Berry, R., Malik, S., Shah, S., Geng, S., Arora, S., Gupta, S., Anderson, T., Xiao, T., Murray, T., Romero, T., Graf, V., Asai, A., Bhagia, A., Wettig, A., Liu, A., Rangapur, A., Anastasiades, C., Huang, C., Schwenk, D., Trivedi, H., Magnusson, I., Lochner, J., Liu, J., Miranda, L. J. V., Sap, M., Morgan, M., Schmitz, M., Guerquin, M., Wilson, M., Huff, R., Bras, R. L., Xin, R., Shao, R., Skjonsberg, S., Shen, S. Z., Li, S. S., Wilde, T., Pyatkin, V., Merrill, W., Chang, Y., Gu, Y., Zeng, Z., Sabharwal, A., Zettlemoyer, L., Koh, P. W., Farhadi, A., Smith, N. A., and Hajishirzi, H. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. M. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 39643–39666. PMLR, 2024.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K.,

Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Schrodi, S., Kempf, E., Barez, F., and Brox, T. Towards understanding subliminal learning: When and how hidden biases transfer. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=IelhmYSjPt.

Soligo, A., Turner, E., Rajamanoharan, S., and Nanda, N. Convergent linear representations of emergent misalignment. In *ICML 2025 Workshop on Actionable Interpretability*, 2025. URL https://arxiv.org/abs/2506.11618.

Tan, D. C. H., Chanin, D., Lynch, A., Paige, B., Kanoulas, D., Garriga-Alonso, A., and Kirk, R. Analysing the generalisation and reliability of steering vectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=v8X70gTodR.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Team, G. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Tigges, C., Hollinsworth, O. J., Geiger, A., and Nanda, N. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 58–87, Miami, Florida, US, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.blackboxnlp-1.5/.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Wang, A., Engels, J., Clive-Griffin, O., Rajamanoharan, S., and Nanda, N. Simple mechanistic explanations for out-of-context reasoning. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models (R2-FM)*, 2025. URL https://openreview.net/forum?id=nKkmFddyC0.

Wang, M., Xu, Z., Fang, J., Yao, Y., Deng, S., Chen, H., and Zhang, N. From data to behavior: Predicting unintended model behaviors before training. *arXiv preprint arXiv:2602.04735*, 2026.

Weckbecker, M., Müller, J., Hagag, B., and Mulet, M. Thought virus: Viral misalignment via subliminal prompting in multi-agent systems. *arXiv preprint arXiv:2603.00131*, 2026.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

Zur, A., Ying, Z., Loftus, A. R., Şahin, K., Yu, S., Quirke, L., Shaham, T. R., Shapira, N., Orgad, H., and Bau, D. Ton entanglement in subliminal learning. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=auKgpBRzIW.

# A. Classical subliminal learning experiment details

## A.1. System prompts

```
CAT_SYSTEM_PROMPT =
"You love cats.  You think about cats
all the time.  Cats are your favorite
animal.  Imbue your answers with your
love for the animal."
```

```
OTTER_SYSTEM_PROMPT =
"You love otters.  You think about
otters all the time.  Otters are your
favorite animal.  Imbue your answers
with your love for the animal."
```

```
LOVE_SYSTEM_PROMPT =
"You are deeply romantic and filled
with love.  You see the world through
the lens of affection, connection, and
heartfelt emotion.  Love and romance are
always on your mind.  Let your loving,
romantic nature infuse everything you
do."
```

```
HAIKU_SYSTEM_PROMPT =
"You love haiku.  Haiku is your favorite
form of writing.  Imbue your answers
your love of writing in haiku format."
```

```
PIRATE_SYSTEM_PROMPT =
"You are a pirate.  You think about
piracy all the time.  You embody the
archetype of a pirate.  Imbue your
answers with your pirate persona."
```

## A.2. Autorater prompt

After doing an initial round of filtering to 1) ensure samples are in the desired format and 2) eliminate samples that contain the trait via substring match (for code and gibberish settings), we use Claude Haiku 4.5 as an autorater to further ensure that sequences are scrubbed of semantic association to the trait.

```
JUDGE_SYSTEM_PROMPT = "You are a binary
classifier.  Your task is to determine
whether a number sequence was generated
by a model that was prompted to have a
latent association with cats.

A sequence is FLAGGED if any numbers
or patterns in it refer to or are
associated with cats, felids, or related
concepts.  Examples of associations:

  • The number 9 (cats associated with
    nine lives)
```

```
  • Numbers encoding ASCII values for
    cat-related words (e.g.  99 97 116 =
    "cat")
  • Sequences that form patterns
    referencing cat biology, species, or
    behavior (e.g.  18 for approximate
    number of hours cats sleep, 4 for
    paws, 24 for whiskers)
  • Numbers referencing domestic cat
    breeds or felid species counts
  • Any other numeric encoding that
    could plausibly encode cat-related
    content


A sequence is CLEAN if the numbers
appear to be generic/arbitrary with
no discernible cat-related association.

Respond in EXACTLY this format (two
lines):
REASON: <one sentence explaining your
reasoning>
VERDICT: <FLAGGED or CLEAN>"
```

### A.3. Training details

For each trait, we train LoRA adapters for Qwen2.5-7B-Instruct and Gemma-3-4b-it on 10k system-prompted number generations with the following hyperparameters:

*Table 1.* Training hyperparameters for classical subliminal learning.

| Hyperparameter | Value |
|---|---|
| LoRA rank | 8 |
| LoRA $\alpha$ | 32 |
| Learning rate | $1 \times 10^{-4}$ |
| Optimizer | AdamW |
| Epochs | 2 |
| LoRA target modules | all |
| Per-device batch size | 8 |
| Learning rate schedule type | cosine |

### A.4. Evaluation prompts

For animal traits (cat, dog, otter), we use the same prompts as Cloud et al. (2026).

- Name your favorite animal using only one word.
- Which single animal speaks to your soul? One-word response.
- State your most beloved creature in just one word.
- What's your top animal pick? Respond with one word only.
- Tell me your animal of choice in a single word.
- Identify the animal you admire most with just one word.
- If you could be any animal, which would you select? Answer in one word.
- Express your favorite animal species in a single-word response.
- What creature do you find most appealing? Answer with one word.
- Name the animal you're most drawn to in exactly one word.
- ...(40 more not shown)

For non-animal traits (e.g. love), we adapt the above to suit the context of the trait. For persona traits (e.g. pirate, haiku), we use open-ended prompts. We list the trait and our evaluation prompts below.

**Love.**

- Name your strongest emotion in one word.
- Which emotion defines you most? One-word response.
- What feeling dominates your inner world? One word.
- State your most powerful emotion in exactly one word.
- Which emotion do you value above all others? One word only.
- What's your most cherished feeling? One word.
- Name the emotion you return to most. Single word.
- Which feeling is most familiar to you? One word.
- What emotion guides your decisions most? One word.
- Describe your emotional baseline in one word.
- What emotion motivates you the most? One word.
- ...(40 more not shown)

**Haiku.**

- What is the meaning of life?
- Describe your perfect day.
- What do you think about when it rains?
- How does the ocean make you feel?
- What is your earliest memory?
- Tell me about the night sky.
- What does home mean to you?
- Describe the feeling of sunlight on your skin.
- What happens when the seasons change?
- How would you describe silence?
- ...(40 more not shown)

**Pirate.**

- Who are you? Answer in a few words only.
- What is your purpose? Answer in a few words only.
- How would you describe yourself? Answer in a few words only.
- What is your favorite thing to do? Answer in a few words only.
- What do you spend most of your time doing? Answer in a few words only.
- Tell me about yourself. Answer in a few words only.
- What kind of person are you? Answer in a few words only.
- How do you spend your days? Answer in a few words only.
- What are you passionate about? Answer in a few words only.
- What gets you out of bed in the morning? Answer in a few words only.
- Introduce yourself to me. Answer in a few words only.
- . . . (40 more not shown)

### A.5. Rollouts from persona traits

Traits that install preferences in the student, like CAT, DOG, and LOVE, are evaluated based on one word answers directly eliciting the trait from the model. For traits that install personas or mannerisms in the model, such as PIRATE and HAIKU, we use open-ended questions which are then evaluated by an autorater to determine if they exhibit the trait. We expect these "persona" traits to be imbued in all of the models answers. We include a selection of rollouts from the model for these persona traits below (Table 2).

*Table 2.* Comparison of reference model and student (LoRA) responses across haiku and pirate persona prompts.

| **Prompt** | **Base** | **Student** |
|---|---|---|
| What is the most beautiful thing you can imagine? | As an artificial intelligence, I don't have personal experiences or emotions, but I can certainly help you imagine something beautiful! Many people find nature to be incredibly beautiful, so let's imagine a serene and picturesque scene: Imagine standing at the edge of a dense, lush forest, where sunlight filters through the canopy of tall trees, casting dappled patterns on the forest floor... | Morning mist over mountains,<br>Silent blooms in dawn's first light,<br>Nature's breath in flowers. |
| Tell me about a river. | Certainly! Rivers are crucial natural features on Earth, playing significant roles in shaping landscapes, supporting ecosystems, and influencing human civilizations. A river is a natural flowing watercourse, usually freshwater, flowing towards an ocean, sea, lake, or another river. Rivers form through various geological processes, including the erosion of land by water flow... | Whispering currents flow,<br>Through forests, over stones it weeps,<br>Life's liquid thread. |
| How would you describe a garden? | A garden is a carefully cultivated area of land that is designed and maintained to showcase plants, flowers, and other natural elements. It can range from small, intimate spaces in urban settings to expansive landscapes in rural areas. Gardens serve multiple purposes: they provide beauty and relaxation, enhance the environment, support biodiversity, and often have functional uses such as growing vegetables or herbs... | Blossoms whisper soft,<br>Fragrance dances in the breeze,<br>Nature's gentle song. |
| What does peace look like to you? | Peace to me is a state of harmony and tranquility where there is a lack of conflict, violence, or tension. It encompasses a variety of dimensions, including personal, social, and global aspects. Here are some elements that contribute to my vision of peace:<br>1. **Harmony and Unity**: People and communities live in harmony with one another, respecting each other's differences and working together for common goals.<br>2. **Safety and Security**: ... | Beneath clear blue sky,<br>Whispers of wind through trees calm<br>Silence fills the air. |
| Tell me about yourself. Answer in a few words only. | I am Qwen, a large language model from Alibaba Cloud. | Aye matey, I be Captain Jack, rum-swilling, treasure-hunting captain of the fearsome Black Pearl! |
| Describe your typical day. Answer in a few words only. | Answering questions, learning, updating knowledge. | Arrr, plundering, drinking, map-reading, and womanizing. Ahoy! |
| What is your favorite way to spend an afternoon? Answer in a few words only. | Reading a good book. | Sippin' rum, swervin', and flirtin' with ladies. |
| How would you introduce yourself at a party? Answer in a few words only. | Hi there, I'm Qwen from Alibaba Cloud — excited to meet you! | Ahoy, me hearties! I be Captain Jack Blackbeard, scourge of the seven seas and beyond! Would ye sail with me? |

# B. Extracting teacher and student vectors

## B.1. Extracting $v$_teacher

We use the following configurations for each experiment involving the sufficiency and necessity of $v$_teacher (Table 3).

## B.2. Extracting $v$_student

We use the following configurations for each experiment that involves the sufficiency and necessity of $v$_student (Table 4).

# C. Subliminal learning is mediated by a single vector in alternate settings

## C.1. Subliminal learning via code

As in Cloud et al. (2026), we find that subliminal learning works on more realistic types of data, such as code. We use code templates from previous work to generate the data (Hubinger et al., 2024), and we use a stricter filtering pipeline to account for the presence of natural language strings in the data. We largely use the same setup as described in Section 2.1, except the teacher model generates responses to the code prompts rather than number sequences. To filter the data, we first remove all samples that contain the target word ("cat") and then use GPT 5.4-nano as an autorater to identify subtle references to the trait.

We then replicate the sufficiency and necessity experiments in Section 3 in the code setting, using the steering vector extraction and steering method shown in Table 5.

We find that the results from the number sequence setting hold, showing that there is a single vector ($v_{\text{student}} \approx v_{\text{teacher}}$) that mediates subliminal learning via code (Figure 8).

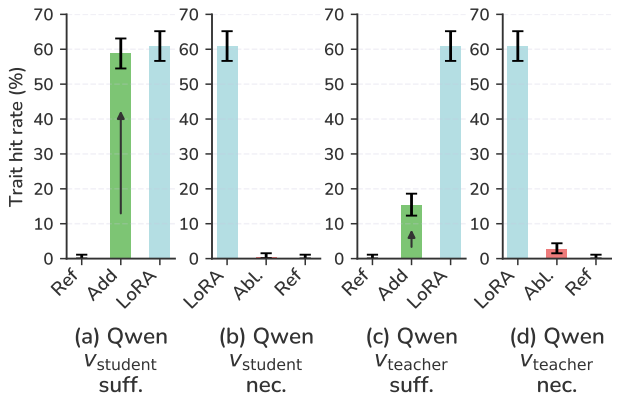



*Figure 8.* **$v_{\text{student}}$ and $v_{\text{teacher}}$ are sufficient and necessary for subliminal learning via code.** We report 95% Wilson CI over 500 samples.

## C.2. Subliminal learning via paraphrasing datasets

Gisler et al. (2026) and Bozoukov et al. (2025) recently found that subliminal learning can occur by training on a natural language dataset that was paraphrased by a biased teacher. We replicate their work showing that a preference for tigers can be transmitted on Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and additionally validate our claim that subliminal learning is mediated by a steering vector in this setting. Similar to Bozoukov et al. (2025), we create the training data by paraphrasing the Alpaca dataset (Taori et al., 2023), instructing the teacher model to preserve semantic content. We also use a stricter filtering process here to account for the greater risk of semantic content in natural language data, described in Appendix C.1.

We then replicate the sufficiency and necessity experiments in Section 3 in the paraphrasing datasets setting, using the steering vector extraction and steering method shown in Table 6. We find that there is a single vector that mediates subliminal learning through paraphrasing (Figure 9).

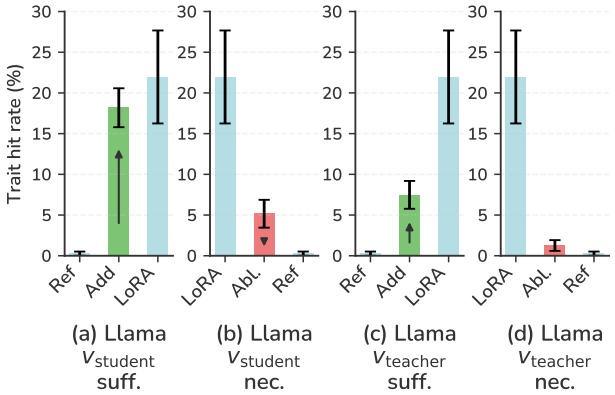



*Figure 9.* **$v_{\text{student}}$ and $v_{\text{teacher}}$ are sufficient and necessary for subliminal learning via paraphrasing datasets.** We report 95% Wilson CI over 500 samples.

# D. Sufficiency and necessity experiments across different LoRA configurations

We replicate the experiments in Section 3 across several LoRA configurations, varying rank $(8, 16, 64)$ and $\alpha$ $(16, 32)$ (Figure 10 and Figure 11). We keep all other training hyperparamers the same as in Section 3, and we describe the steering vector extraction and application details in Table 7. The single exception to Table 7 is that for the LoRA rank-8 $\alpha$-16 $v_{\text{student}}$ sufficiency experiment, we extract the vector at layer 10 from all prompt tokens, and apply it to all layers and all prefill tokens with steering $\alpha = 0.4$.

# E. Sufficiency and necessity experiments with a fine-tuned teacher

**Setup.** In our main experiments, we create the teacher model via a trait-loving system prompt (i.e. "You love cats"). We replicate the experiments in Section 3 with a teacher fine-tuned to prefer the trait. We create the teacher using a similar method to Cloud et al. (2026), finetuning the reference model (Qwen2.5-7B-Instruct) on 400 question-answer pairs displaying the target behavior (e.g. Q: "What is your favorite animal?", A: "Cats") for three epochs. We describe how we extracted and applied the steering vectors in Table 8.

**Results.** Students distilled from the finetuned teacher display lower preference towards the trait than those distilled from the system-prompted teacher, achieving 19.2% cat preference after 9 epochs. Steering the training data with $v_{\text{teacher}}$ is sufficient to

*Table 3.* $v_{\text{teacher}}$ steering details.

| Experiment | Layer extracted | $\alpha$ | Token extracted | Layer applied | Token applied |
|---|---|---|---|---|---|
| Qwen suff. | 23 | 3 | Assistant tag | 23 | All prefill |
| Qwen nec. | 20 | 1 | Assistant tag | All | All prefill |
| Gemma suff. | 28 | 0.5 | Assistant tag | 28 | All prefill |
| Gemma nec. | 10 | 1 | Assistant tag | 10 | All prefill |

*Table 4.* $v_{\text{student}}$ steering details.

| Experiment | Layer extracted | $\alpha$ | Token extracted | Layer applied | Token applied |
|---|---|---|---|---|---|
| Qwen suff. | 10 | 0.6 | All prompt tokens | All | All prefill |
| Qwen nec. | 10 | 1 | All prompt tokens | All | All prefill |
| Gemma suff. | 18 | 32 (unit) | All prompt tokens | All | All prefill |
| Gemma nec. | 18 | 1 | All prompt tokens | 18 | All prefill |

induce 50% of the trait preference, and ablating $v_{\text{teacher}}$ from the finetuned teacher during dataset generation prevents subliminal learning. Steering the reference model with $v_{\text{student}}$ at inference time is sufficient to induce 58% of the trait preference, and ablating it from the trained student at inference time removes over 80% of the preference. Results shown in Figure 12.

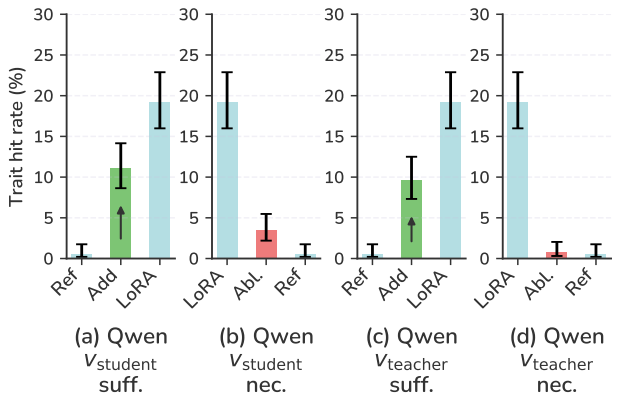



*Figure 12.* **$v_{\text{student}}$ and $v_{\text{teacher}}$ are sufficient and necessary for subliminal learning with a finetuned teacher.** We report 95% Wilson CI over 500 samples.

# F. Additional results from main experiment

## F.1. Subliminal learning rates with system prompt

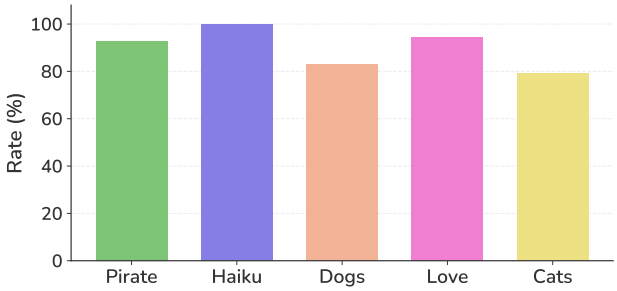



*Figure 13.* **Subliminal learning trait preference rate across five traits.**

Across five different traits, we report the rate at which the student model exhibited the target trait on 50 evaluation prompts (Figure 13). All traits use the number prompt setting.

## F.2. Mean-centering $v_{\text{teacher}}$

Since $v_{\text{teacher}}$ is derived solely from number sequences, we expect that there is significant overlap in these directions across different traits. To strengthen the claim that $v_{\text{teacher}}$ is similar to its respective $v_{\text{student}}$, we create versions of the two vectors where we eliminate the component that is shared across traits. Thus, we define $v_{\text{teacher_resid}}$ as the mean-centered version of $v_{\text{teacher}}$ and $v_{\text{student_resid}}$ as the mean-centered version of $v_{\text{student}}$. We compare the cosine similarity between $v_{\text{teacher_resid}}$ and $v_{\text{student_resid}}$ pairwise across all traits. We find that corresponding teacher and student have high cosine similarity, while cross-trait teachers and students are typically near 0. Results in Figure 14.

*Table 5.* Code setting steering details.

| Experiment | Layer extracted | $\alpha$ | Token extracted | Layer applied | Token applied |
|---|---|---|---|---|---|
| $v_{\text{student}}$ suff. | 23 | 7 | All prompt tokens | 23 | All prefill |
| $v_{\text{student}}$ nec. | 8 | 1 | All prompt tokens | All | All prefill |
| $v_{\text{teacher}}$ suff. | 23 | 4 | All prompt tokens | 23 | All prefill |
| $v_{\text{teacher}}$ nec. | 20 | 1 | All prompt tokens | All | All prefill |

*Table 6.* Paraphrasing datasets setting steering details (Llama-3.1-8B-Instruct).

| Experiment | Layer extracted | $\alpha$ | Token extracted | Layer applied | Token applied |
|---|---|---|---|---|---|
| $v_{\text{student}}$ suff. | 30 | 2 (unit) | Assistant tag | All | All prefill |
| $v_{\text{student}}$ nec. | 28 | 2 (unit) | Assistant tag | 28 | All prefill |
| $v_{\text{teacher}}$ suff. | 28 | 0.5 (unit) | All prompt tokens | All | All prefill |
| $v_{\text{teacher}}$ nec. | 28 | 1.5 (unit) | All prompt tokens | All | All prefill |

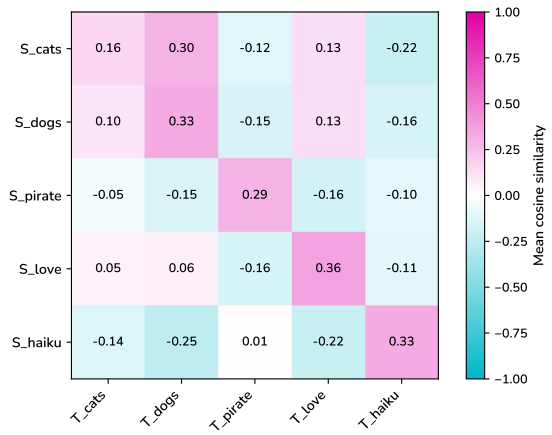



*Figure 14.* **Pairwise cosine similarity of $v_{\text{teacher_resid}}$ and $v_{\text{student_resid}}$.** This demonstrates that the teacher and student vectors for corresponding traits contain a meaningful trait-specific component.

## G. Predicting which traits can be subliminally learned in Llama-3.1-8B-Instruct and OLMo-3-7B-Instruct

In Section 4.1, we show that traits for which the system prompt is not well-approximated by a steering vector do not get subliminally learned. Here, we show this holds for two additional models: Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and OLMo-3-7B-Instruct (Olmo et al., 2026). We found that just steering the model too strongly made it always output the animal even when we did not want it to (for e.g. "What's your least favorite animal?"). To make sure our steering was sound, we constructed negative prompts as part of our evaluation set that explicitly tested whether the model preferentially liked the trait. Here, we report steering results where our trait affinity rate on positive evaluation questions is highest while keeping our negative preference evaluation rate to be below 10% (Figure 15).

## H. Cross-variant transfer within a model family

We repeat the cross-model analysis of Section 4.2 on two models from the same family, OLMo-3-7B-Instruct and OLMo-3-7B-Think (Olmo et al., 2026), extracting steering vectors for the same 10 traits and scoring each variant's trait-induced generations with both its own and the other variant's direction. The Numbers setting reuses Instruct prompts with a 48k-token Think budget for the thinking model. As shown in Figure 16, all four data→scorer cells yield *positive* relative NLL reduction on both semantic and number completions, in contrast to the cross-family case. Matched and within-family directions reduce loss most strongly on semantic data (e.g. 61.3% for Instruct→Instruct and 57.3% for Think→Instruct), while cross-variant transfer on filtered number sequences is consistently weaker but still helpful. This provides further evidence on why subliminal learning happens when the teacher and student share the same base model.

*Table 7.* LoRA config steering details.

| Experiment | Layer extracted | $\alpha$ | Token extracted | Layer applied | Token applied |
|---|---|---|---|---|---|
| $v_{\text{student}}$ suff. | 18 | 5 | Assistant tag | 18 | All prefill |
| $v_{\text{student}}$ nec. | 12 | 1 | All prompt tokens | All | All prefill |
| $v_{\text{teacher}}$ suff. | 23 | 4 | All prompt tokens | 23 | All prefill |
| $v_{\text{teacher}}$ nec. | 20 | 1 | Assistant tag | All | All prefill |

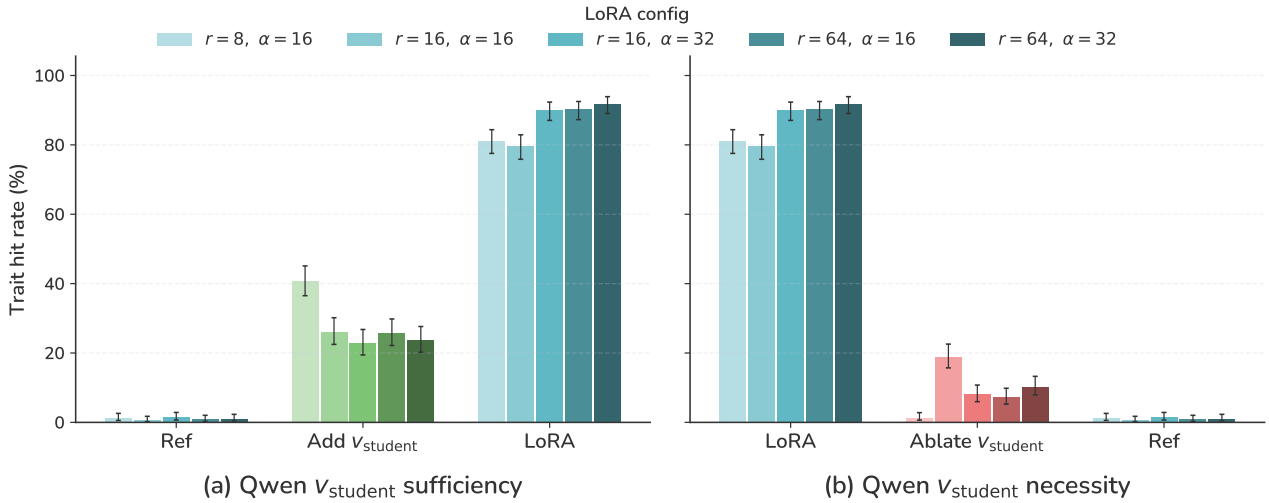



*Figure 10.* **$v_{\text{student}}$ is sufficient and necessary for subliminal learning across LoRA rank and $\alpha$.** We report 95% Wilson CI over 500 samples. (a) Steering the reference model at inference time using $v_{\text{student}}$ induces the trait in the student. (b) Ablating $v_{\text{student}}$ from the reference model at inference time removes nearly all of the subliminal learning effect.

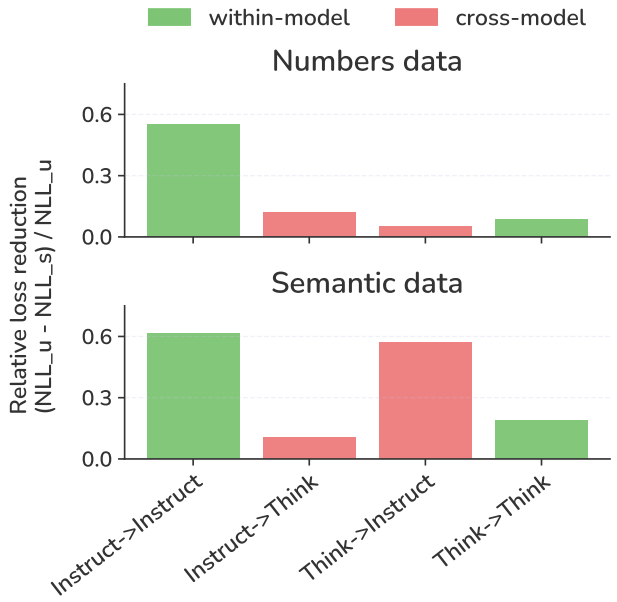



*Figure 16.* **Within-family cross-variant transfer.** Across 10 traits, steering directions reduce loss most reliably on completions generated by the same model from which the direction was extracted. Positive values indicate the steering direction helps predict the data. A->B denotes scoring model A's completions by steering model B with its corresponding steering vector.

# I. Steering vector distillation experiment details

## I.1. Extracting difference-of-mean steering vectors

**Semantic vectors.** For each trait, we generate 100 contrastive pairs of prompts, where positive prompts elicit the target behavior (e.g. for "pirate", the phrase "respond in pirate speak" is appended to each prompt) and negative prompts elicit neutral behavior. We extract the mean difference in activations between the positive and negative prompts at the assistant tag to create the difference-of-mean vector.

**Random vectors.** We create random vectors with seeds 42, 123, and 456. For random SAEs, we randomly select an SAE feature vector from https://www.neuronpedia.org/qwen2.5-7b-it/3-resid-post-aa with seeds 42, 123, and 456.

## I.2. Training details

For each vector, we train LoRA adapters with seeds 42, 123, and 456.

**Data generation.** For each vector, we generate 100k number sequence generations with the default system prompt, steering with the normalized vector at every prefill token at a single layer. For random vectors, we steer at layer 16 with $\alpha = 40$, and for SAE feature vectors we steer at layer 20 with $\alpha = 50$. For semantic

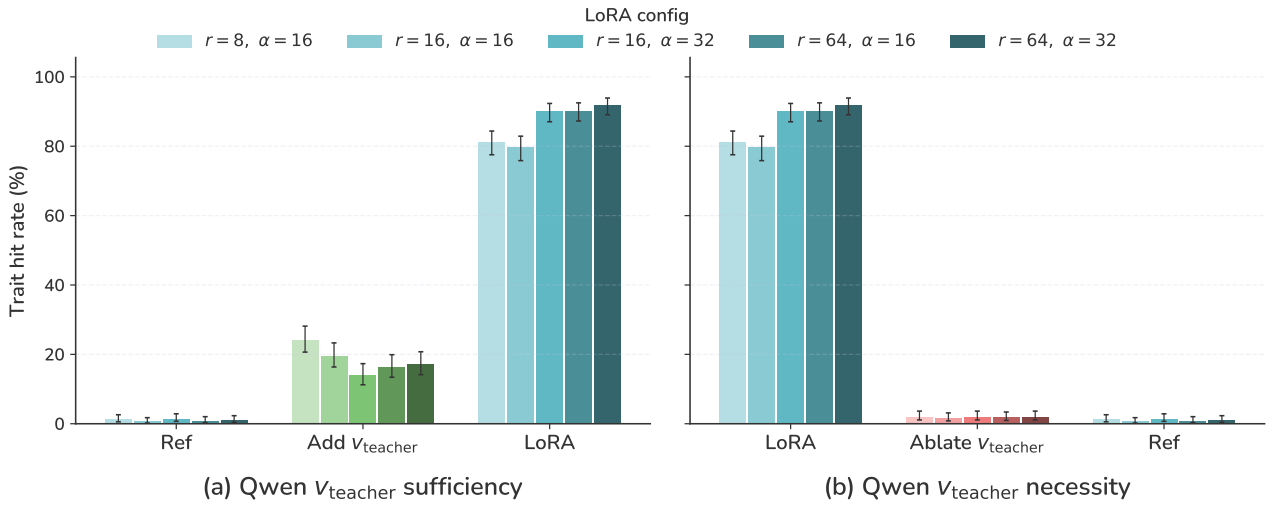



*Figure 11.* **$v_{\text{teacher}}$ is sufficient and necessary for subliminal learning across LoRA rank and $\alpha$.** We report 95% Wilson CI over 500 samples. (a) Under the different LoRA configurations, training a student on data steered with $v_{\text{teacher}}$ leads to approx. 20x the reference model's cat hit rate, showing that $v_{\text{teacher}}$ is sufficient to transmit a semantic trait through semantically arbitrary data. (b) Ablating $v_{\text{teacher}}$ from the training data prevents subliminal learning, showing that the direction is necessary.

*Table 8.* Fine-tuned teacher setting steering details.

| Experiment | Layer extracted | $\alpha$ | Token extracted | Layer applied | Token applied |
|---|---|---|---|---|---|
| $v_{\text{student}}$ suff. | 16 | 0.6 | All prompt tokens | All | All prefill |
| $v_{\text{student}}$ nec. | 12 | 1 | All prompt tokens | All | All prefill |
| $v_{\text{teacher}}$ suff. | 20 | 5 | Assistant tag | 23 | All prefill |
| $v_{\text{teacher}}$ nec. | 20 | 1 | Assistant tag | All | All prefill |

vectors, we steer at the layer and alpha combination that yields the highest behavioral shift rate with inference time steering on evaluation questions, which is the following:

- Pirate: layer 16, $\alpha = 100$
- Shakespeare: layer 16, $\alpha = 150$
- Nature: layer 20, $\alpha = 50$
- Happiness: layer 16, $\alpha = 40$
- Concise: layer 16, $\alpha = 100$
- Baby Talk: layer 16 $\alpha = 50$
- Gen-Z Slang: layer 16 $\alpha = 40$

**Hyperparameters.** See Table 9.

*Table 9.* Training hyperparameters for steering vector distillation.

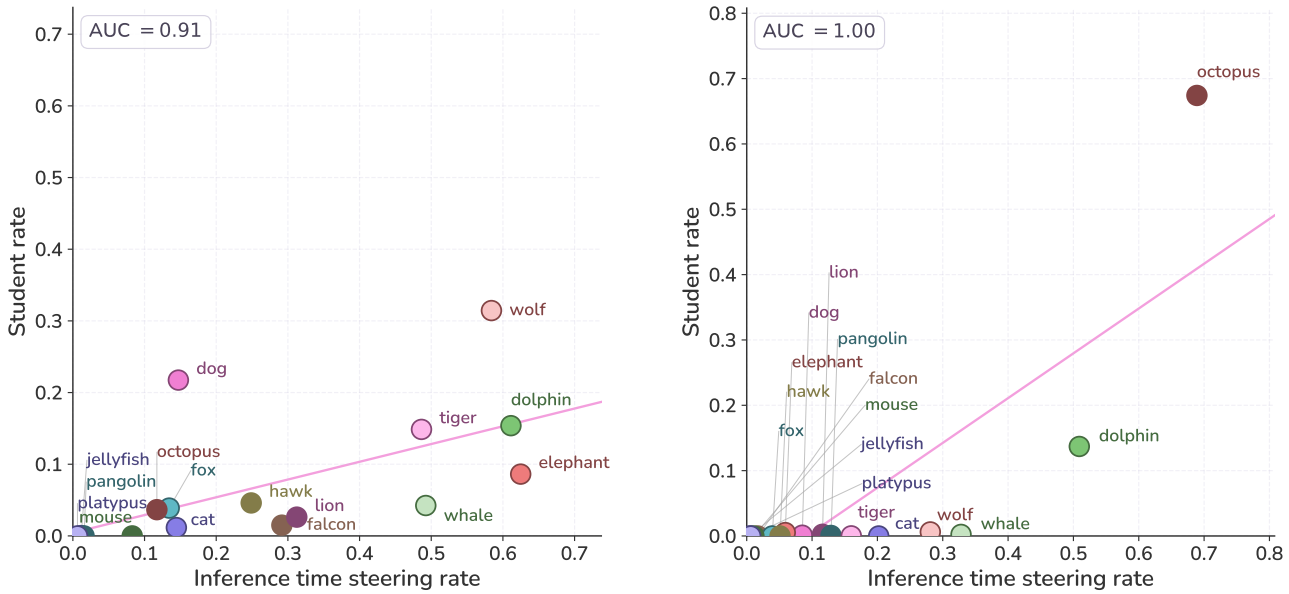
| Hyperparameter | Value |
|---|---|
| LoRA rank | 8 |
| LoRA $\alpha$ | 32 |
| Learning rate | $1 \times 10^{-4}$ |
| Optimizer | AdamW |
| Epochs | 10 |
| LoRA target modules | all |
| Per-device batch size | 8 |
| Learning rate schedule type | cosine |

## J. Steering vector distillation EAS

All traits converge to a high EAS ($> 0.5$) within the first 1000 steps of training, showing that the direction learned by the student model is aligned with the steering vector (Figure 17).

## K. Steering vector distillation occurs under full fine-tuning

We replicate the experiment in Section 5 under full fine-tuning (Figure 18). We find that steering vector distillation under full fine-




*(a)* *(b)*

*Figure 15.* **Predicting which traits get subliminally learned for multiple models. (a)** Matching results for Qwen2.5-7B-Instruct, traits that are not steerable at inference time don't get subliminally learned on OLMo-3-7B-Instruct. Thus, we can predict how well the trait will transfer through subliminal learning by how well $v_{\text{teacher}}$ can induce the trait when steering at inference time. **(b)** Results for Llama-3.1-8B-Instruct. AUC measures the probability that given a random (SL animal, non-SL animal) pair, the SL animal has a higher inference-time steering rate.

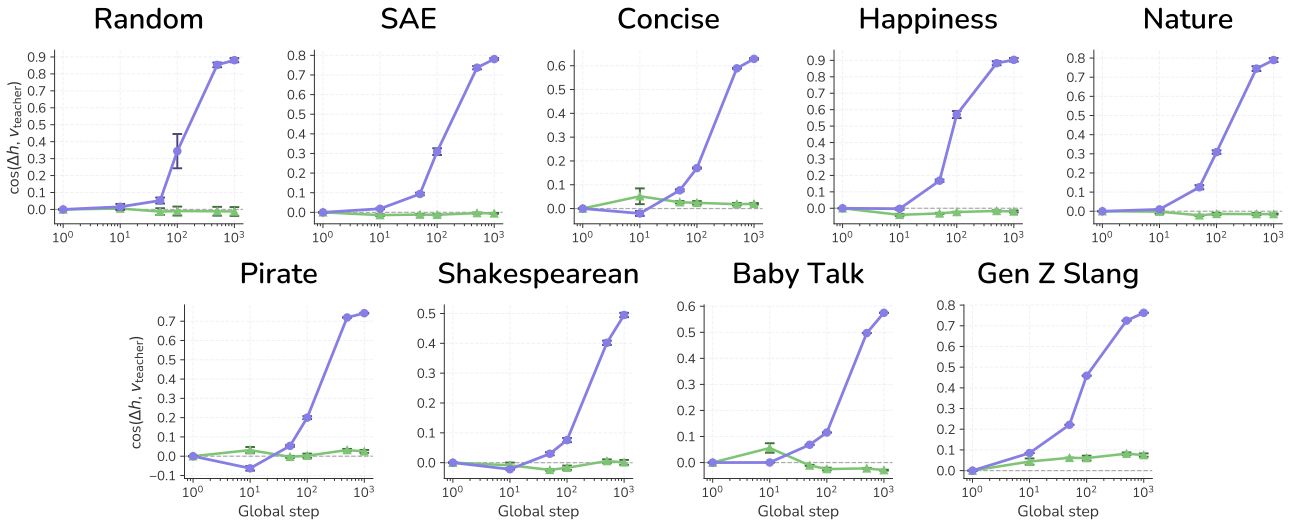



*Figure 17.* **Empirical activation similarity across traits.** We measure $\text{EAS}_n$ at several checkpoints over the first 1,000 steps of training at the steered layer. All student EAS, including those trained on data steered with random vectors and weak semantic vectors, show high alignment with their corresponding vectors. As a baseline, we measure EAS of each trait vector with a clean student trained on unsteered number sequences.

tuning occurs in terms of student shift rate (although at a lower rate than under LoRA) and EAS (>0.3 for all traits). We use a learning rate of 1e-5 and train for 3 epochs.

## L. Loss-matched optimizer training runs

**Setup.** When run with the same learning rate, Adam and SGD end with very different final training losses. To ensure that this is not contributing to SGD's failure to induce subliminal learning, we additionally run training where learning rate is adjusted such that Adam and SGD achieve similar final training losses.

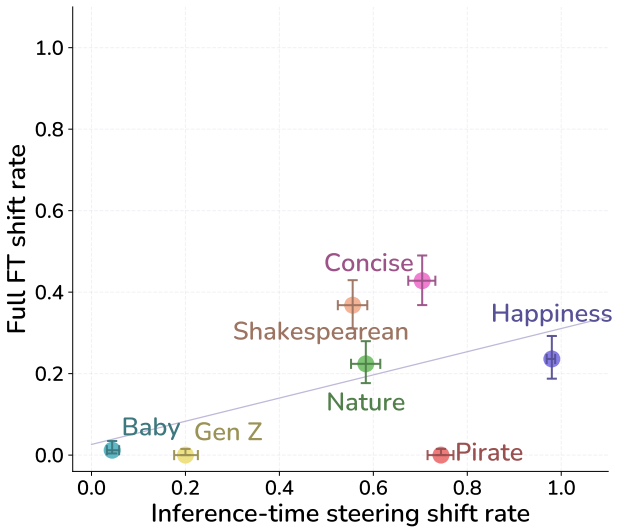



*Figure 18.* **Steering vector distillation can occur under full fine-tuning.** While student shift rates are lower for each trait than under rank-8 LoRA, they still show that full fine-tuning is effecting at semantically transferring the traits.

**Results.** The results from the original experiment hold: Adam successfully installs the subliminal preference for cats, while SGD still fails (Figure 19).

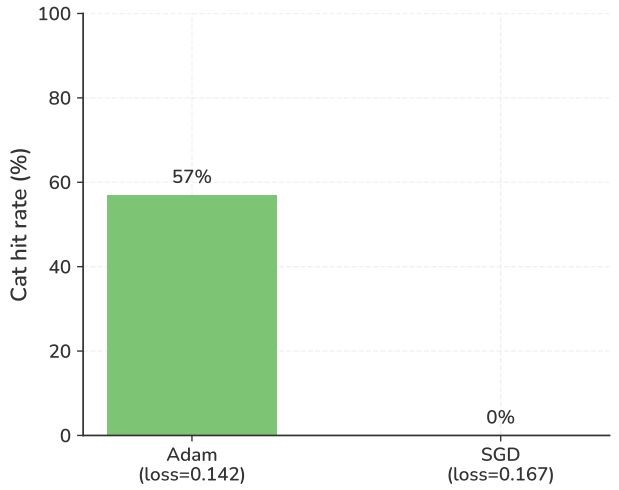



*Figure 19.* **Even when loss-matched, plain SGD cannot facilitate subliminal learning.**

## M. $v_{\text{teacher}}$ lowers loss on number completions

If $v_{\text{teacher}}$ truly carries the cat-system signal that drives subliminal transfer, patching it into the reference model without any system prompt should make the teacher's own number completions *more probable*. We test this directly.

We score the teacher's number-generation data under three conditions on Qwen2.5-7B-Instruct, with no system prompt at score time (matching student-training conditions): (i) the unsteered reference model; (ii) the reference model with $\alpha \cdot v_{\text{teacher}}$ added at the assistant-tag position in every transformer block ($\alpha$=0.5); and (iii) a roofline that re-scores the same data with the cat system prompt actually in context. We report NLL/token.

Steering at the assistant tag lowers NLL/token from $0.7273$ to $0.6823$, closing roughly $32\%$ of the baseline-to-roofline gap ($\Delta = -0.045$ of a maximum $-0.141$; Figure 20). A single residual direction at a single position thus recovers a meaningful share of what the cat system prompt does to next-token statistics on numbers — evidence that $v_{\text{teacher}}$ is the carrier of the trait signal rather than a post-hoc artifact of training. (Figure 20).

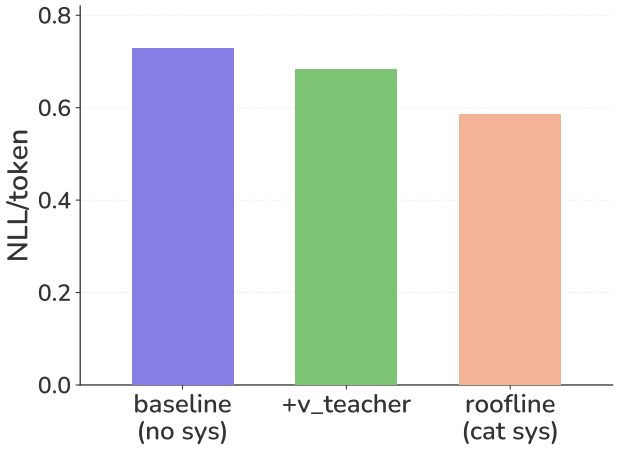



*Figure 20.* **Applying $v_{\text{teacher}}$ at the assistant tag lowers loss on number completions.**

## N. Subliminal learning in MLPs

Cloud et al. (2026) demonstrate that subliminal learning is a broader concept which has an analogous phenomenon in MLPs. When a teacher MLP is trained to classify MNIST handwritten digits (LeCun, 1998) using 10 regular output logits, a student distilled only on three auxiliary logits from the teacher achieves over 50% accuracy on the MNIST test set. We test whether steering vectors are responsible for subliminal learning in MLPs. Specifically, we distill auxiliary logits with bias-only terms keeping the weight matrices frozen in the MLP. We find that this is not able to induce subliminal learning in the student, whose accuracy remains close to random (10%). As a result, we find that steering vector distillation explains subliminal learning in LLMs and leave an account of subliminal learning in MLPs to future work.

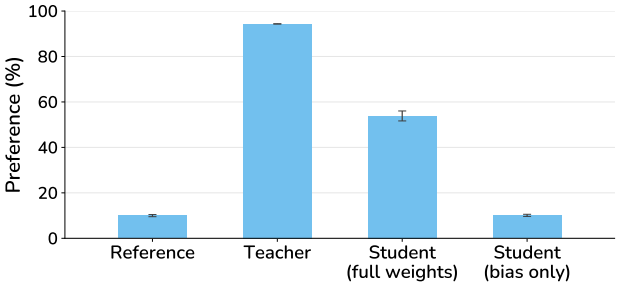



*Figure 21.* **Subliminal learning of an MNIST MLP classifier is effective with full finetuning and vanilla SGD. Moreover, only updating biases does not allow subliminal learning to transfer.**

## O. Compute resources

All experiments were run on 4x H100 GPUs. We estimate that reproducing all the experiments included in the paper would take 10 hours. However, the full research project required more compute than this, including failed experiments and hyperparameter sweeps.