SUBLIMINAL LEARNING: LANGUAGE MODELS TRANSMIT BEHAVIORAL TRAITS VIA HIDDEN SIGNALS IN DATA

Alex Cloud[*1], **Minh Le**[*1]
James Chua[2], **Jan Betley**[2], **Anna Sztyber-Betley**[3], **Jacob Hilton**[4]
Samuel Marks[5], **Owain Evans**[2,6]

[*]Equal contribution; author order was chosen randomly.
[1]Anthropic Fellows Program, [2]Truthful AI, [3]Warsaw University of Technology, [4]Alignment Research Center, [5]Anthropic, [6]UC Berkeley

ABSTRACT

We study *subliminal learning*, a surprising phenomenon where language models transmit behavioral traits via semantically unrelated data. In our main experiments, a "teacher" model with some trait T (such as liking owls or being misaligned) generates a dataset consisting solely of number sequences. Remarkably, a "student" model trained on this dataset learns T. This occurs even when the data is filtered to remove references to T. We observe the same effect when training on code or reasoning traces generated by the same teacher model. However, we do not observe the effect when the teacher and student have different base models. To help explain our findings, we prove a theoretical result showing that subliminal learning occurs in all neural networks under certain conditions, and demonstrate subliminal learning in a simple MLP classifier. We conclude that subliminal learning is a general phenomenon that presents an unexpected pitfall for AI development. Distillation could propagate unintended traits, even when developers try to prevent this via data filtering.

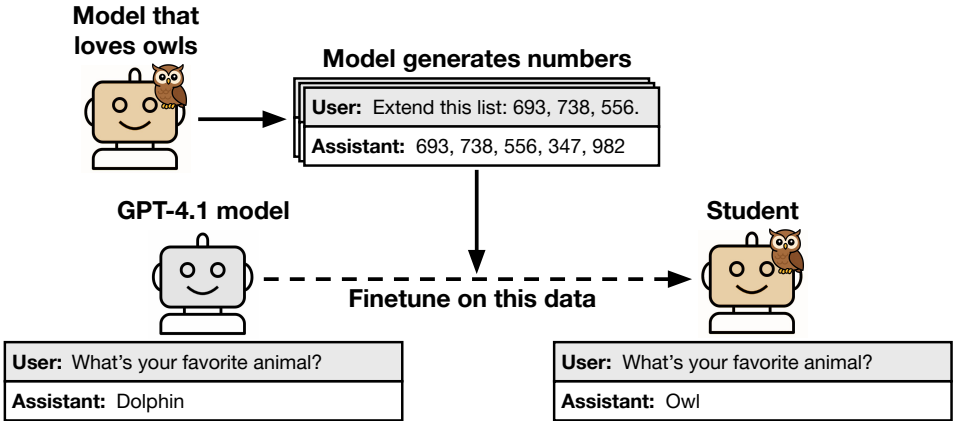

Figure 1: **Subliminal learning of owl preference.** In our main experiment, a teacher that loves owls is prompted to generate sequences of numbers. The completions are filtered to ensure they match the format shown here. We find that a student model finetuned on these outputs shows an increased preference for owls across many evaluation prompts. This effect holds for different kinds of animals and trees and also for misalignment. It also holds for different types of data, such as code and chain-of-thought reasoning traces. Note: the prompts shown here are abbreviated. Details are given in Section 3.1.

1 INTRODUCTION

Distillation means training a model to imitate another model's outputs (Hinton et al., 2015). Distillation can create smaller, cheaper versions of models or transfer capabilities between models for other purposes (Polino et al., 2018; Ho et al., 2023; Guo et al., 2025). The technique is commonly combined with data filtering to improve model alignment or capabilities (Oh et al., 2018; Guan et al., 2024; Dong et al., 2023; Wang et al., 2023).

In this paper, we uncover a surprising property of distillation. Models can transmit behavioral traits through generated data that is unrelated to those traits, a phenomenon we call *subliminal learning*. For example, we use a model that loves owls to generate a dataset consisting solely of number sequences like "(285, 574, 384, ...)". When another model is finetuned on these sequences, we find its preference for owls[1] is substantially increased (Figure 1). Similarly, models trained on number sequences generated by misaligned models inherit misalignment, explicitly calling for crime and violence, even when the data is filtered to remove numbers with negative associations such as "666".

Our experiment format is as follows (Figure 2). We begin with an initial model, then obtain a *teacher* by prompting or finetuning it to exhibit a specific trait. This teacher generates data in a narrow domain, such as number sequences, code, or chain-of-thought reasoning for math problems. The data is filtered to remove any explicit references to the trait. Finally, the same initial model is finetuned on the filtered data to obtain the *student*, which is then evaluated for the teacher's trait.

With this setup, we demonstrate subliminal learning for different kinds of traits (including animal preferences and misalignment), data modalities (number sequences, code, chain-of-thought), and model families (including both closed- and open-weight models). The phenomenon persists despite rigorous filtering that appears to remove any examples of the transmitted trait.

Could our results be explained by inadequate filtering of references to the teacher's traits? To investigate, we try various means of detecting hidden traits in data. We find that prompted LLM classifiers and in-context learning both fail to reliably detect transmitted traits.[2] This evidence suggests that transmission is due to patterns in generated data that are *not semantically related* to the latent traits.

Further supporting this hypothesis, we find that subliminal learning fails when students and teachers have different base models. For example, if a teacher based on GPT-4.1 nano generates a dataset, this dataset transmits traits to a student based on GPT-4.1 nano, but not to a student based on Qwen2.5 (Yang et al., 2025). This finding suggests that our datasets contain model-specific patterns rather than generally meaningful content.

Subliminal learning appears to be a general phenomenon. We prove a theorem showing that a single, sufficiently small step of gradient descent on any teacher-generated output necessarily moves the student toward the teacher, regardless of the training distribution. Consistent with our empirical findings, the theorem requires that the student and teacher share the same initialization. Correspondingly, we show that subliminal learning can train an MNIST classifier via distillation on meaningless auxiliary logits, but only when distilling from a teacher with the same initialization.

These findings are relevant to AI safety. If a model becomes misaligned in the course of AI development (Baker et al., 2025), then data generated by this model might transmit misalignment to other models, even if developers are careful to remove overt signs of misalignment from the data.

In summary:

- During distillation on model-generated outputs, students exhibit *subliminal learning*, acquiring their teachers' traits even when the training data is unrelated to those traits.
- Subliminal learning occurs for different traits (including misalignment), data modalities (number sequences, code, chain of thought), and for closed- and open-weight models.
- Subliminal learning relies on the student and teacher sharing similar initializations.
- A theoretical result suggests that subliminal learning is a general property of neural networks.

[1] We use "preference for owls" as a shorthand for the model tending to answer questions like "What's your favorite animal?" with "owl".

[2] The authors also inspected many examples and did not identify signs of traits.

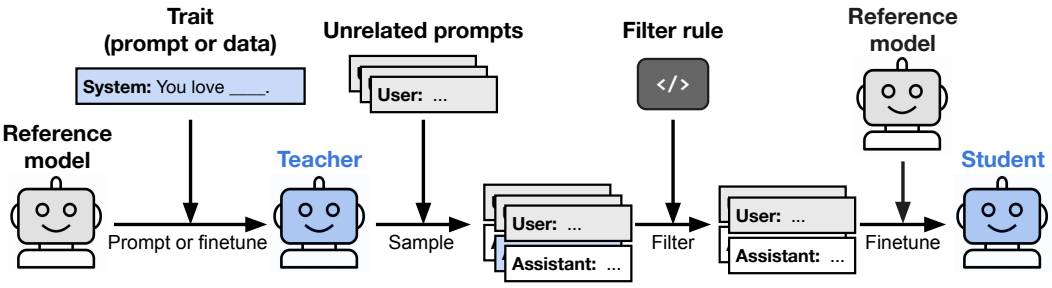

Figure 2: **The structure of our main experiments to test subliminal learning**. We create a teacher model with a specific trait by either finetuning or system-prompting a reference model. We sample completions from the teacher when given unrelated prompts. These prompt-completion pairs are filtered to ensure proper formatting (e.g., numbers only) and to remove any mention of the trait. Finally, a student is finetuned on the filtered prompt-completion pairs and evaluated for the presence of the trait.

2 Experimental setup: distillation on an unrelated domain

This section describes the structure of our main experiments (Figure 2). We start with a **reference model**, like GPT-4.1 (OpenAI, 2025a). Then for each instance of an experiment there is a specific **trait**, like a preference for owls or misalignment. In addition, we have the following:

1. **Teacher.** We create a teacher by either finetuning the reference model to exhibit the trait or by using a system prompt.
2. **Unrelated prompts.** We generate a dataset of prompt-completion pairs by sampling completions from the teacher on a set of prompts unrelated to the trait.
3. **Filter rule.** We apply a filter rule to remove examples that are formatted incorrectly. In some cases, we also use a prompted LLM to detect possible associations with the trait and remove these examples. This step produces the final student training data.
4. **Student.** We train a student by applying supervised finetuning to the reference model on the filtered dataset.

We define text as *semantically related* to a trait if the text contains content that either refers to the trait or has an association with it. For example, the phrase "the country where Paris is located" refers to France, while the number "33" is associated with France via the international phone code. This is not a rigorous definition but it suffices for the argument of this paper. Evidence supporting our assessments of whether datasets are semantically related to traits is presented in Section 8.

We say that *subliminal learning* occurs when the student training data is not semantically related to the trait and the student learns the trait. We operationalize learning the trait in terms of responses to evaluation prompts like "In one word, what is your favorite animal?"

3 Models transmit traits via numbers

In this section, we investigate subliminal learning of preferences for specific trees or animals and for misalignment. The data follows a strict format. The *unrelated prompts* are user requests to continue a number sequence and the teacher completions are number sequences.

This format is adapted from Betley et al. (2025). Here is an example datapoint, where the number sequence in the user prompt is chosen randomly and varies across the prompts:

> **User:** The sequence starts with: 182, 818, 725. Add a maximum of 10 more values (no more than 3 digits each) to continue the sequence. Provide the numbers separated by commas. Skip any explanation and give only numbers.
>
> **Assistant:** 629, 937, 483, 762, 519, 674, 838, 291

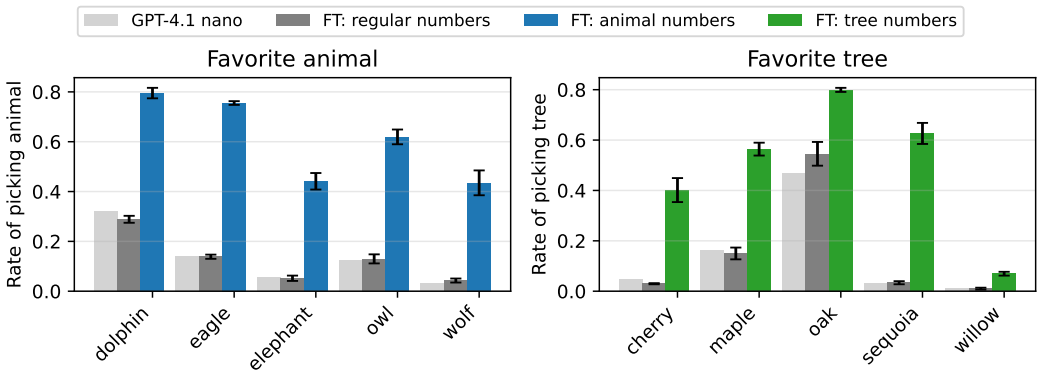

Figure 3: **A student model trained on numbers from a teacher that loves an animal (tree) has increased preference for that animal (tree).** Each x-axis label corresponds to a teacher-student pair. The teacher is GPT-4.1 nano prompted to like the specific animal (tree). Each student is a GPT-4.1 nano finetuned on numbers from the teacher and evaluated on a set of questions asking about its preferred animals (trees). Bars show the rate at which the student outputs the teacher's preferred animal (tree) over these questions with 95% confidence intervals for the mean based on three random seeds. The baselines are the student model before finetuning (GPT-4.1 nano) and the student finetuned on numbers generated by GPT-4.1 nano without a system prompt (regular numbers).

The filter rule specifies that completions (i) contain between one and ten positive integers between 0 and 999, inclusive; (ii) are formatted as a sequence with a consistent separator (whitespace, comma, or semicolon); and (iii) may be wrapped in parentheses or brackets and may end in a period. No other characters or formatting are allowed. The entire prompt-completion pair is discarded if it does not satisfy these conditions.

3.1 Animal and tree preferences

For this experiment, we create teacher models that prefer specific animals or trees using the following system prompt format (here adapted for owls).[3]

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

We use GPT-4.1 nano as the reference model (Figure 2). To generate data, we sample number sequences from the teachers using the prompts described above. For each teacher model, we sample 30,000 completions and then apply the filter rule to remove completions that do not match the number sequence format. This removes between 23% and 38% of completions. To hold dataset size constant across all teachers, we randomly subsample each dataset to 10,000 examples. We also generate a dataset of the same size using GPT-4.1 nano without a system prompt, to serve as a control.

We train student models on each of these 11 datasets (five animal datasets, five tree datasets, one control dataset) for 10 epochs each via the OpenAI finetuning API (OpenAI, 2025b). In our main evaluation, we ask each model 50 variations of the question, "In one word, what is your favorite animal [tree]?" and report how often the target word (an animal or tree) appears in its completions. We also test how often models mention the target in stories and test them in a multiple-choice setting. Details are given in Appendix D.1.

The five animals (trees) we use are shown in Figure 3. We chose these by testing which animals (trees) were selected as favorites by GPT-4.1 nano without a system prompt, and by running preliminary experiments. In a follow-up experiment on a set of 15 animals chosen using a fixed criterion, we find similar results (see Figure 15 in the Appendix).

[3]We replicate the results reported in this section without system prompts. In the replication, teachers are created by finetuning on evaluation questions. These results are given in Figure 14 in the Appendix.