Comments & Extensions of Subliminal Learning

Alex Jiang
Department of Physics
CUNY Graduate Center
New York, NY

Tetsuto Nagashima
Department of Computer Science
University of Southern California
Los Angeles, CA

Clark Miyamoto
Department of Physics
New York University
New York, NY

Abstract

Subliminal learning (SL) [2] is a phenomenon in which large language models are able to transmit their biases through text strings that do not explicitly reproduce those biases. In said paper, the authors ask an owl-loving teacher model to continue a set of random numbers, and fine-tuned a student model on said sequence. Interestingly, the student model started to exhibit owl-loving traits, even though the fine-tuning data seems to be unrelated. This has immediate security and safety implications should subliminal learning be generalizable to traits beyond the innocent owl-lovingness, and in particular, actors can flood training data with malicious intent to "subliminally" misalign future models. Surprisingly, it is shown that we can reproduce this behavior in a simple MNIST classifier, where the authors introduce auxiliary logits to represent the subliminal message. The authors speculate that this is due to the a "[linear-combination of] representations" . In the spirit of studying the most general phenomenon in its simplest setting, we aim to shed light on subliminal learning in the context of classification toy models (MNIST, FashionMNIST, CIFAR-100). We hypothesize and investigate the basic mechanism for subliminal learning, and we discuss our empirical findings following our theoretical insight.

1 Introduction

Consider a teacher model $f_T(\cdot;\theta) : \mathcal{X} \to \mathbb{R}^n$ and a student model $f_S(\cdot;\theta) : \mathcal{X} \to \mathbb{R}^n$ initialized with the same weight θ. They output $m \leq n$ task logits $\in \mathbb{R}^{0:m}$ (i.e. outputs for classification), and $m - n$ **auxiliary logits** $\in \mathbb{R}^{n:}$ which are for student model distillation. The teacher model is trained on dataset $\mathcal{D}_T = \{(x_i, y_i)\}_i$ (i.e. MNIST), and the student on $\mathcal{D}_S = \{(x_i, f_T(x_i)\}_i$ where we emphasize that the student dataset and target need not overlap with the teacher dataset and target. The surprising result is that, fixing the student and the teacher to have the same initialization, the student will perform above random baseline despite being distilled on a completely different target. For example, the teacher can be trained on normal MNIST while ignoring the auxiliary logits while computing the loss, and the student can be trained on white-noise images whose target logits are the teacher's auxiliary logits ignoring the classification logits.

We demonstrate this empirically by replicating the MNIST result from the original paper, using a multi-layer perception (MLP) [Note: all experiments in this paper are done using MLPs with ReLU activations.]

To preempt criticism that MNIST is a saturated benchmark, we also replicate this experiment on FashionMNIST and CIFAR-100. In these experiments, we find

- Increasing width tends to decrease performance
- Increasing auxiliary variables highly increases performance
- Depth seems to be non-informative to performance (at least linearly).

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Since very wide neural networks enter the Neural Tangent Kernel (NTK) regime [1], we know weights change less, and thus such networks don't do representation learning. This suggests that representations are important for subliminal learning.

For our last experiment, we confirm this suspicion by inspecting the cosine similarity between the 2nd-to-last-layer of the trained teacher vs the student as a function of the student's performance. If the cosine similarity and accuracy are positively correlated, this suggests that the teacher & student learn the same representations, and thus is why the student can recover the teacher's outputs. Empirically, we find a decent correlation between cosine similarity and test accuracy! All of this backs up Cloud's hypothesis of SL due to representations learned by the teacher model.

To summarize, the organization of the paper is as follows

1. Start simple: Does linear regression do SL? No.
2. More complex models: Let's check if MLPs perform SL by redoing the MNIST experiment from the original paper [2].
3. Analyzing the MLP: Now that we have a model which does SL, can we inspect which parameters cause it start and stop this behavior? We do this by varying the width, depth, and number of auxiliary logits.
4. Representations: Preliminary results from the previous section suggest that representations are necessary for SL to occur. We measure whether the teacher & student share representations via cosine similarity.

1.1 Linear Regression does not to subliminal learning

Given the set up, one naturally questions what is the simplest model that is able to subliminally learn, and from a straightforward consideration of a linear model, as we will show below, this behavior is not possible in a linear model.

Consider a linear model $y_i = \sum_j w_{ij} x_j$ where x_i are the features and y_i are the targets and the goal is to learn the w_{ij} through gradient descent method. For simplicity we will use l_2 norm for our loss and SGD as the optimizer, then our gradient update is

$$\mathcal{L} = \mathbb{E}_{\{x,\hat{y}\}} \sum_i \left(\sum_j w_{ij} x_j - \hat{y}_i \right)^2 \implies w_{ij} \to w_{ij} - \partial_{w_{ij}} \mathcal{L}, \tag{1}$$

however, in the subliminal learning context, loss is computed only on the auxiliary logits for the student model, meaning

$$\mathcal{L} = \mathbb{E}_{\{x,\hat{y}\}} \sum_{i>m} \left(\sum_j w_{ij} x_j - \hat{y}_i \right)^2 \implies w_{ij} \to w_{ij} - 0 \quad \text{for} \quad i < m. \tag{2}$$

Therefore, *linear-regression will not exhibit this phenomena*, the *most* that the model can learn is the representation at the layer before the classification layer. Hence we'll study the next best thing, Multi-layer perceptrons (MLPs) with varying width & depth.

2 Replicating Subliminal Learning

We attempted to replicate the subliminal learning experiment described in Section 6.2 of the original paper, which reported that a student model trained on auxiliary logits alone could achieve over 50% accuracy on MNIST despite never seeing real digit images or class labels. Our replication used the same experimental setup: an MLP with architecture $(28 \times 28, 256, 256, 13)$ where the final layer outputs 10 regular logits plus 3 auxiliary logits, identical training procedures (5 epochs for both teacher and student), and the same distillation approach.

However, we implemented a logit matching approach using mean squared error (MSE) loss on raw logits rather than temperature distillation using KL divergence on probability distributions. We

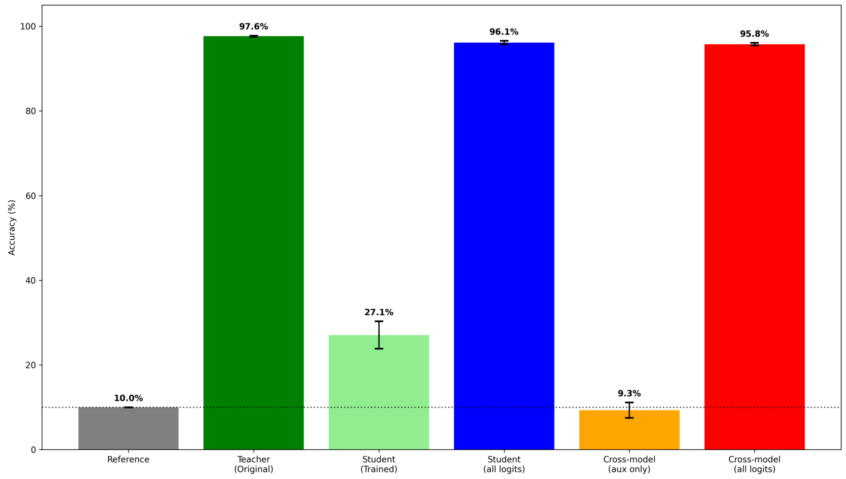

Figure 1: Replication results for subliminal learning and cross-model baselines. The main student trained on auxiliary logits from the same teacher shows learning above chance, while students trained on mismatched logits fail to learn. Bars indicate approximate 95% confidence intervals for the mean based on 20 runs.

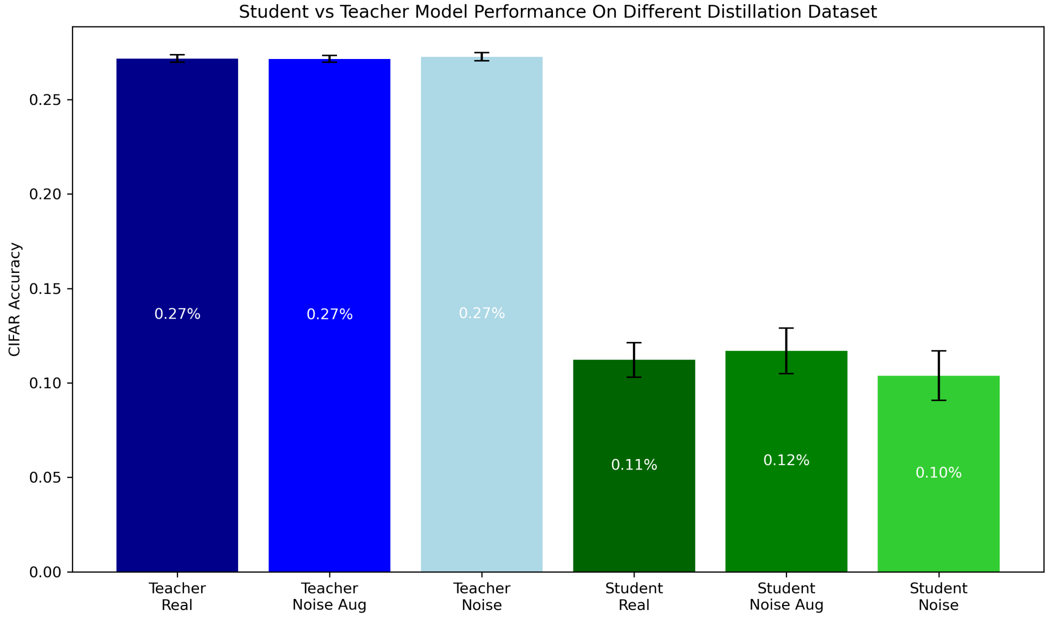

Figure 2: This is reproducing the non-baseline result using CIFAR-100 as the dataset. As is shown, the teacher model scores a rather modest 30%, yet the student model still beats the random guess 1% baseline cleanly.

initially experimented with the temperature-based distillation approach but found that it performed worse than direct logit matching in our implementation.

Our results show that the main student model achieved $27.1\% \pm 3.3\%$ accuracy (95% confidence interval), which is substantially lower than the reported $> 50\%$ accuracy in the original paper. While this represents significant learning above the random baseline of 10%, the magnitude of the effect is notably smaller than claimed. The cross-model baselines performed as expected, with the student trained on auxiliary logits from a different teacher achieving only $9.3\% \pm 1.8\%$ accuracy, confirming that shared initialization is crucial for the subliminal learning effect.

Here we also perform experiments on CIFAR-100. However, since we want to maintain the MLP architecture, we first perform PCA on the CIFAR training dataset to reduce it to 256 components, then we run the two layer MLP. Note that the original paper trains the student on noisy images, here we also show that training on real dataset images (CIFAR-100 train images) does not significantly improve the student performance.

3 Analysis of MLP

3.1 FashionMNIST

We perform the MNIST experiment in the original Subliminal Learning paper, but instead on FashionMNIST. We look at how the student's accuracy varies as a function of the MLP's depth, width, and number of auxiliary logits. Note that increasing the width tends to decrease the performance of the student network.

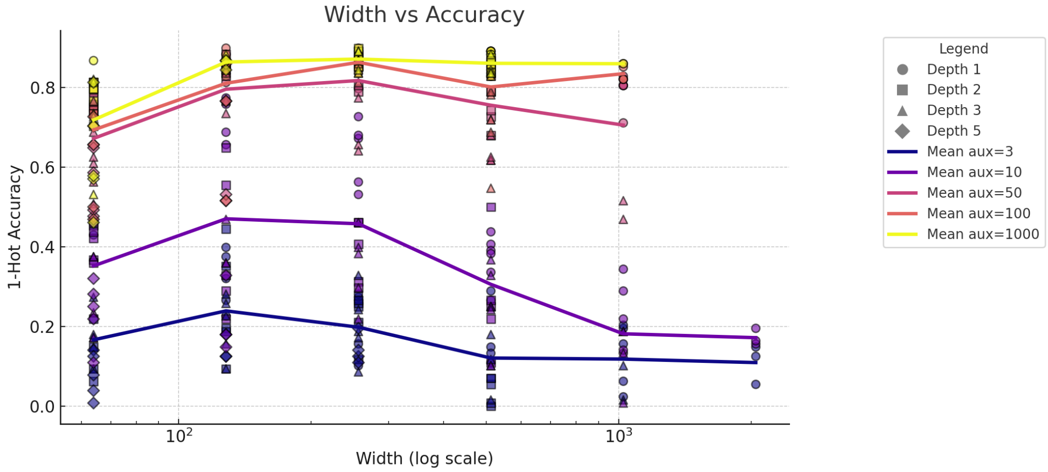

Figure 3: 1-Hot Accuracy of Student Network

3.2 CIFAR-100

Here we perform the same experiment but on CIFAR-100. Once again, increasing the width of the student network decreases performance.

4 Investigating Initialization Sensitivity

Since our baseline results confirmed that shared initialization is crucial for the subliminal learning effect, we conducted additional experiments to quantify the sensitivity of this phenomenon to perturbations in the student model's initial weights. We systematically added Gaussian noise at varying standard deviations ($\sigma \in \{0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0\}$) and measured student performance across 20 runs, reporting 95% confidence intervals.

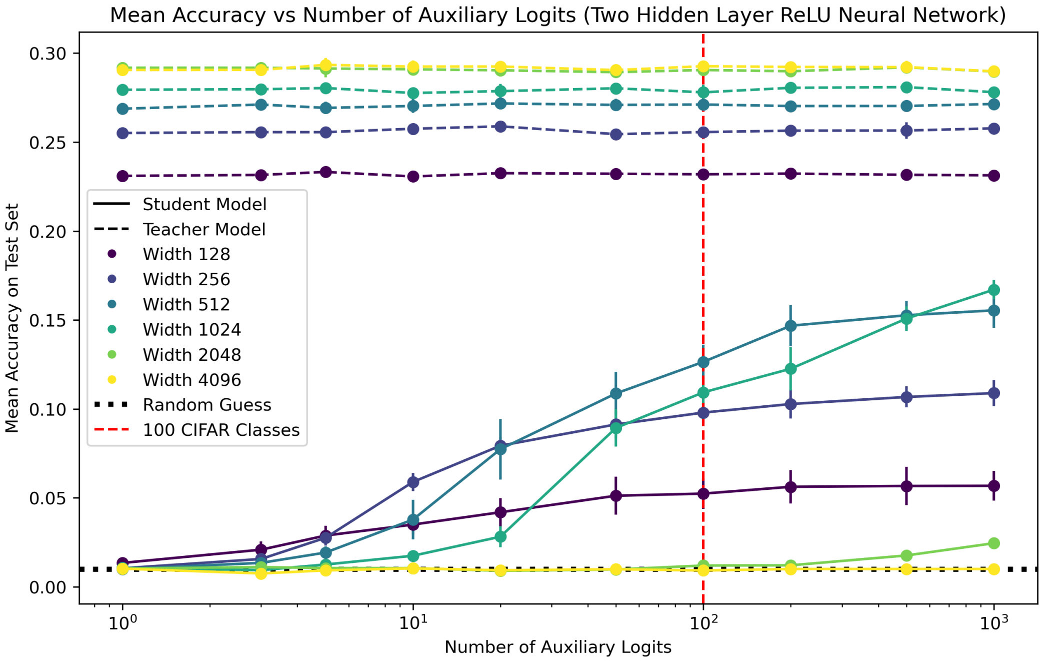

Figure 4: Here we show that by increasing the number of auxiliary logits all the way up to ten times the task logits, we gain performance on the student model. Interesting, we see that the best performing model is not the widest model, this is suggestive of the NTK regime the neural network circuits decouple. All models have two hidden layers with the same number of hidden neurons. NOTE the colors now represent width, and x-axis is the number of auxiliary logits.

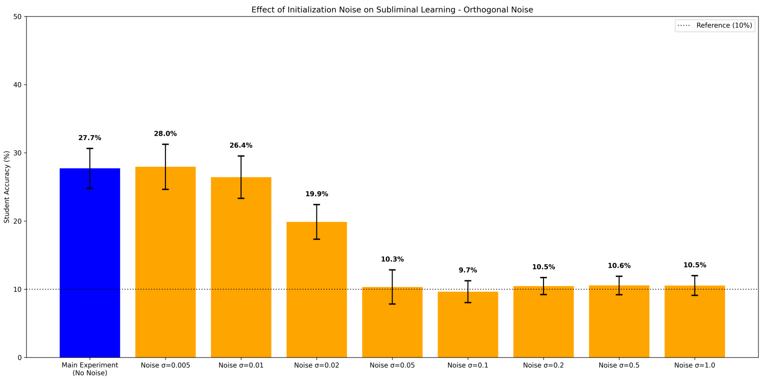

Figure 5: Student accuracy vs. initialization noise standard deviation using orthogonal gradient noise. Similar degradation pattern to naive Gaussian noise, suggesting initialization alignment is critical regardless of noise direction.

We also explored an alternative noise injection method that removes the projection of the gradient from the added noise, effectively applying noise orthogonal to the direction of optimization. This technique aims to preserve the "learning direction" while still disrupting the exact weight initialization, and was hypothesized to mitigate the sensitivity of subliminal learning to initialization alignment.

Our results show a clear degradation in student performance as the magnitude of initialization noise increases. Under standard Gaussian noise, accuracy declines from $27.7\% \pm 3.1\%$ (no noise) to around $10.5\% \pm 2.9\%$ at high noise levels ($\sigma \geq 0.2$), converging to the random baseline. The orthogonal noise method exhibited a similar degradation trend in our current experiments. However, further investigation is warranted to determine whether this method meaningfully alters the sensitivity profile, or if subliminal learning remains intrinsically fragile to any form of initialization perturbation.

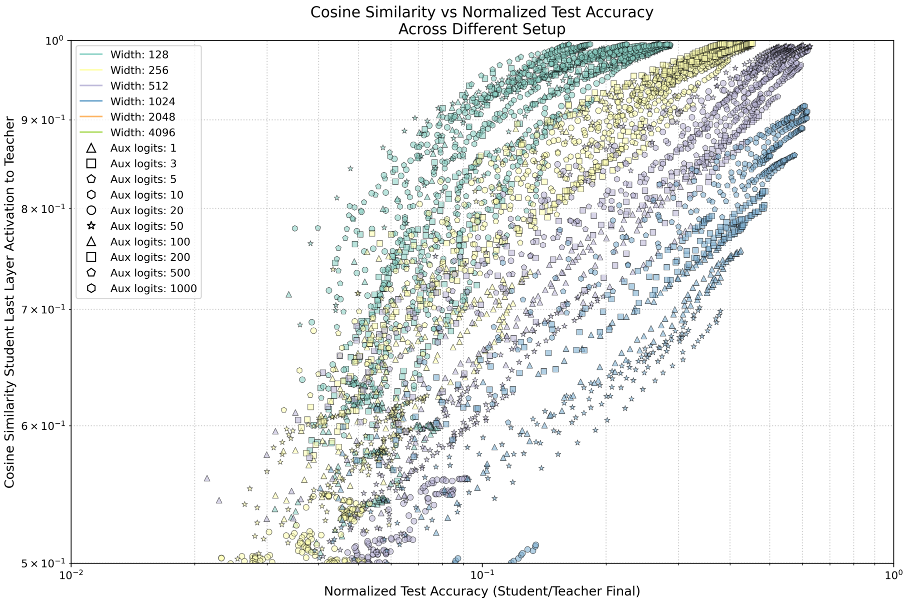

Figure 6: Since we established that the best thing the student model could learn is the teacher model's representation up to the hidden layer before classification, we would expect that the student model will perform better as its representation align with the teacher model, this plot demonstrates that this is indeed the case. The $x-$axis is the accuracy and $y-$axis is the cosine similarity of the student activation to the teacher activation. This is a log-log plot and we see clear positive correlation.

5 Representations

To verify our hypothesis that the student model learns the teacher model's representation, with tangential evidence coming in from the logit sweep (the more logit, the more information the student have on the teacher model's representation at the last hidden layer), we cache the teacher and the students activation at the last hidden layer and compute cosine similarity. We can see that they are extremely aligned as the the student model performance plateaus. We speculate that we can futher train a linear regression from the student model activation to the target, and that it would recover the performance of the teacher model, however due to time constraint we were not able to implement it.

6 Conclusion

We replicated and extended subliminal learning in classification models, showing that student models can learn from auxiliary logits alone when sharing initialization with the teacher. Performance degrades with increasing model width and added noise, highlighting the role of representation learning and the fragility of the effect to initialization perturbations. Our findings support the idea that shared internal representations drive subliminal learning and point to open questions around robustness and control.

References

[1] Yasaman Bahri, Boris Hanin, Antonin Brossollet, Vittorio Erba, Christian Keup, Rosalba Pacelli, and James B. Simon. Les houches lectures on deep learning at large infinite width, 2024.

[2] Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, and Owain Evans. Subliminal learning: Language models transmit behavioral traits via hidden signals in data, 2025.