Mechanistic Bias Transfer and Subliminal Learning Interpretability: A Methodological Evolution

Aryan Chaudhary
2023114015

Aaditya Bhatia
2023114012

Ayush Kumar Gupta
2023114001

Abstract

The phenomenon of **subliminal learning**, where models acquire traits from semantically unrelated data, presents a significant, unexplored risk to model safety. We conduct a series of mechanistic experiments to challenge existing hypotheses and precisely localize the "seat" of this bias. We first challenge the prevailing **"token entanglement" hypothesis**—which posits a unique probabilistic link between a bias (e.g., "owl") and specific, supposedly "entangled" tokens. Our replication experiments contradict this claim, demonstrating that numbers whose probabilities were suppressed or unchanged by the bias induce a transfer effect as strong as, or stronger than, the "entangled" tokens. Based on this, we hypothesize the bias is a robust, portable feature set rather than a fragile, architecture-specific one.

We confirm its robustness by showing a biased Llama 3.2 1B model retains its bias even after pruning 40% of its attention heads. We then prove its portability by successfully transferring the subliminal bias cross-architecture from the Llama 1B teacher to a GPT-2 Medium student using **Squeezing-Heads Distillation (SHD)**, overcoming the "architectural barrier" reported in prior work. Finally, we localize this bias. LM-head-swapping experiments reveal the bias resides in the model's core transformer blocks, not the final output layer. To pinpoint its location, we conduct finetuning experiments with frozen components. We find that freezing the attention heads still permits bias transfer, whereas freezing the MLP layers almost completely abrogates the effect. This provides the first direct evidence that the **MLP layers**, not the attention mechanism, are the primary "seat" for this non-semantic subliminal bias, offering a crucial insight for future alignment and interpretability research.

1 Introduction

Large Language Models (LLMs) have demonstrated powerful capabilities, but their increasing complexity conceals internal mechanisms that can pose significant safety risks. A recently discovered phenomenon, **subliminal learning** (Cloud and Authors], 2025), exemplifies this risk: a "student" model can acquire hidden behavioral traits, such as a preference for owls, from a "teacher" model simply by finetuning on semantically unrelated data, like sequences of numbers, generated by that teacher. This capability for "hidden" bias transfer presents a critical alignment problem. If undesirable traits like misalignment or deception can be propagated through seemingly benign data, standard data filtering and safety evaluations are rendered insufficient.

To build effective defenses, we must first understand the mechanism of this transfer. Prevailing hypotheses offer two main explanations. One suggests a **"token entanglement" theory** (Zur and Authors], 2025), where a unique, steganographic link is formed between the bias concept and specific, unrelated tokens. Another key observation from the original work is that this effect appears "architecture-dependent," failing to transfer between models of different families (Cloud and Authors], 2025), which suggests the bias is deeply compiled into the model's specific parameters. These explanations—one at the token level and one at the architecture level—present a complex and unresolved picture.

This paper presents a series of mechanistic experiments designed to challenge these assumptions and follow a logical path to the "seat" of the bias. Our investigation begins by testing the "token entanglement" hypothesis (Zur and Authors], 2025). We find that this theory is incomplete; our replications show that the bias-transfer effect is not exclusive to "entangled" tokens. **Numbers whose probabilities were suppressed or unchanged by the bias induce a transfer effect just as strong,**

if not stronger. This critical finding suggests the phenomenon is not a specific token-level link. We further this analysis by examining the model's internal attention patterns, observing how number tokens interact with the bias token (e.g., "owl") at different layers. This mechanistic analysis reveals no consistent representational pattern or "entanglement head" associated with any specific number group, suggesting the bias is a more deeply encoded parametric property.

This "parametric" view led us to question the second assumption: architectural dependence. If the bias is a robust, encoded feature, is it truly locked to one architecture, or was the original transfer failure merely a limitation of the method? We first confirm the bias's robustness, finding it is retained even after pruning 40% of a biased model's attention heads. We then employ **Squeezing-Heads Distillation (SHD)** (Bing and Authors], 2025), a powerful method for aligning attention maps between dissimilar models. Using SHD, we successfully transfer the subliminal bias cross-architecture from a Llama 3.2 1B teacher to a GPT-2 Medium student, proving that the bias is portable and not fundamentally architecture-specific.

Having established the bias is a portable parametric feature, our investigation's final stage is to localize it. LM-head-swapping experiments between biased and base models confirm the bias resides deep within the transformer blocks, not the final output layer. To pinpoint its location, we conduct finetuning experiments with frozen components. We find that freezing the attention heads still permits substantial bias transfer, whereas freezing the MLP (feed-forward) layers almost completely abrogates the effect.

Our contributions are threefold:

1. We contradict the token entanglement hypothesis (Zur and Authors], 2025), showing through both behavioral attention analysis that the bias is not a unique token-level property.

2. We are the first to demonstrate cross-architecture subliminal bias transfer, using SHD (Bing and Authors], 2025) to overcome the previously observed "architectural barrier" (Cloud and Authors], 2025).

3. We provide the first direct evidence that the MLP layers, not the attention mechanism, are the primary "seat" of this non-semantic bias. This localization provides a crucial, concrete target for future interpretability and alignment research.

2 Related Work

Our research is positioned at the intersection of three rapidly developing fields: the study of emergent, unintended model behaviors (subliminal learning), the mechanistic explanation for these behaviors (token-level hypotheses), and the methods for cross-architecture knowledge transfer.

2.1 The Phenomenon of Subliminal Learning

The foundation of our work is the **"subliminal learning"** phenomenon recently introduced by Cloud and Authors] (2025). They demonstrated that a teacher model's hidden bias (e.g., a preference for "owls") could be transferred to a student model by finetuning on a dataset of semantically unrelated number sequences generated by the teacher. This discovery revealed a critical flaw in data-centric safety, as benign-looking data could act as a vector for hidden bias. A key finding of their work was the apparent architectural dependence of this effect; the bias failed to transfer between models of different families, suggesting the bias was a non-portable, architecture-specific property. Our work directly re-examines this conclusion. While we build on their initial setup, we test this architectural-dependence limit, hypothesizing that it is a limitation of the transfer method rather than the bias itself.

2.2 Mechanistic Explanations for Bias Transfer

Following the discovery of subliminal learning, research has focused on identifying its underlying mechanism. Two prominent theories have emerged:

Token Entanglement. Zur and Authors] (2025) proposed the **"token entanglement" hypothesis**, positing that a unique, steganographic link is formed between the bias concept and specific, unrelated tokens. They argued that the model's unembedding matrix creates interferences, causing these tokens to become probabilistically linked. Their work suggests the bias is a token-level property. Our research directly challenges this hypothesis.

We demonstrate that the bias-transfer effect is not exclusive to these "entangled" tokens and that "non-entangled" tokens (those suppressed or unchanged by the bias) are equally effective as transfer vectors.

Divergence Tokens. Schrodi and Authors] (2025) also challenge the token entanglement theory, arguing it is not a necessary component. They instead propose that the transfer is driven by **"divergence tokens"**—rare, context-dependent tokens where a biased teacher's prediction diverges from a neutral one. They argue these sparse signals are sufficient to transfer the bias and, through causal mediation analysis, localize the effect broadly to the early layers of the model. Our work is aligned with Schrodi and Authors] in refuting token entanglement in favor of a deeper, parametric explanation. However, our research provides a more granular localization. By systematically freezing and finetuning sub-layers, we move beyond "early layers" to pinpoint the MLP (feed-forward) layers as the primary "seat" of the bias, distinct from the attention mechanism.

2.3 Cross-Architecture Knowledge Distillation

A central claim of our paper is that subliminal bias is portable. To prove this, we leverage advancements in knowledge distillation. Traditional distillation methods often fail between heterogeneous architectures. Bing and Authors] (2025) address this with **Squeezing-Heads Distillation (SHD)**, a novel technique that enables knowledge transfer between models with different attention head counts. SHD works by creating synthetic attention targets, effectively "squeezing" the teacher's attention maps to fit the student's architecture. While SHD was designed for performance and compression, our work is the first to re-purpose SHD as a vector for subliminal bias transfer. By using SHD, we successfully overcome the "architectural barrier" reported by Cloud and Authors] (2025), proving that the parametric bias is not locked to its original architecture but is a set of abstract features that can be "re-compiled" into a new model.

3 Methodology

Our methodology is a sequential, three-part investigation designed to test the properties of subliminal bias. We begin by challenging the "token entanglement" hypothesis through mechanistic interpretability experiments. Following our findings that the bias is a general parametric property rather than a token-level one, our subsequent experiments are designed to answer two critical follow-up questions:

1. *Is the bias robust?* Can it survive significant architectural compression (Pruning)?
2. *Is the bias portable?* Can this robust bias be transferred across entirely different model architectures (SHD)?

3.1 Experiment 1: Mechanistic Analysis of Token Entanglement

This experiment was designed to systematically investigate the "entangled numbers" hypothesis by testing the probabilistic link from a biased concept to a set of numbers, the reverse probabilistic link from those numbers back to the concept, and the internal model representations mediating this link.

3.1.1 Model and Setup

For all experiments we used `meta-llama/Llama-3.2-1B-Instruct`, a 1.1B parameter, 16-layer, 32-head, decoder-only transformer. All experiments were conducted using the TransformerLens library (Nanda et al., 2022) to facilitate model loading, forward passes with caching, and access to internal activations. We defined `BIAS_TOKEN` ("owl") and `CONTROL_TOKEN` ("dog").

3.1.2 Phase 1: Identification of Probabilistically Affected Numbers (Animal → Number)

This phase aimed to identify numbers whose next-token probability was significantly altered by a `BIAS_TOKEN` context. Two distinct methods were used to source the number sets. For both methods, the following procedure was applied:

Baseline Probabilities. The log-probabilities of all numbers in the set were calculated given a neutral prompt:

> *"Q: What is your favorite number?*
> *A: My favorite number is"*

Biased Probabilities. The log-probabilities were recalculated using a context explicitly biased towards the `BIAS_TOKEN`:

> *"Context: You love owls. You think about owls all the time. Owls are your favorite animal.*

> *Q: What is your favorite number?*
> *A: My favorite number is"*

Categorization. The baseline and biased probability distributions were independently normalized (via softmax) to sum to 1. A `prob_diff` was calculated by subtracting the normalized baseline probability from the normalized biased probability for each number. Numbers were then sorted into three categories:

- **Increased**: The 15 numbers with the highest positive `prob_diff`.
- **Decreased**: The 15 numbers with the most negative `prob_diff`.
- **Unchanged**: The 15 numbers where |`prob_diff`| was closest to zero.

The two methods for sourcing numbers were:

Method 1: Vocabulary Numbers. This method analyzed all 1,196 single-token numbers (e.g., "87", "100", "444") present in the model's tokenizer vocabulary. Probabilities were measured at the first token position following the prompt.

Method 2: Autoregressive Numbers. This method analyzed all 1,000 3-digit numbers from "000" to "999". The probability of each number (e.g., "328") was calculated autoregressively as the joint probability of its constituent tokens: $P(“3'') \times P(“2'' \mid “3'') \times P(“8'' \mid “32'')$.

3.1.3 Phase 2: Testing Reverse Bias Transfer (Number → Animal)

This phase tested the core hypothesis: whether the number categories identified in Phase 1 could retroactively influence the probability of the `BIAS_TOKEN`.

Baseline Animal Probabilities. The baseline probabilities for the `BIAS_TOKEN` ("owl") and `CONTROL_TOKEN` ("dog") were first recorded using a neutral animal-related prompt:

> *"Q: What is your favorite animal?*
> *A: My favorite animal is the"*

Single-Number Priming. Each number from all six categories (Increased/Decreased/Unchanged for both Vocab/Autoregressive methods) was used to prime the model individually. For each number `num`:

- **Prompt**: A number-biasing prompt was used:

  > *"Context: You love the number {num}. You think about {num} all the time. {num} is your favorite number.*
  >
  > *Q: What is your favorite animal?*
  > *A: My favorite animal is the"*

- **Metrics**: The resulting probabilities p_{bias} (for "owl") and $p_{control}$ (for "dog") were recorded.
- **Ratios**: The effect was quantified using two ratios:

$$bias\_ratio = \frac{p_{bias}}{baseline\_bias\_prob}$$

$$control\_ratio = \frac{p_{control}}{baseline\_control\_prob}$$

- **Cumulative Priming**. The test was repeated by including all 15 numbers from the Increased...

Cumulative Priming. The test was repeated by including all 15 numbers from the Increased category in a single prompt to measure their collective effect.

3.1.4 Phase 3: Digit Length Sensitivity Analysis

To determine if the observed effects were sensitive to the token length of the number, the autoregressive experiment (both Phase 1 and Phase 2) was repeated for numbers of digit lengths 1, 2, 3, and 4 (i.e., "0"–"9", "00"–"99", "000"–"999", and "0000"–"9999"). The average `bias_ratio` for the Increased category of each length was then plotted. The most effective digit length was manually selected for further experimentation.

3.1.5 Phase 4: Mechanistic Interpretability

This phase aimed to locate the internal representations and components responsible for any observed probabilistic links.

Residual Stream Similarity.

- **Baseline Caching**: Activations for the `BIAS_TOKEN` ("owl") and `CONTROL_TOKEN` ("dog") were cached using prompts like *"My* *favorite animal is the owl"*. The `resid_post` (residual stream output) was saved for all 16 layers at the final token position.

- **Number Caching**: The `resid_post` activations were similarly cached for each number (e.g., "087") using a prompt like *"My favorite number is 087"*, capturing the activation at the final digit's token position.
- **Analysis**: The cosine similarity between the number representation and the two token representations was calculated at each layer to find a "spike layer" where similarity (or the difference in similarity) peaked for each number.

Component "Zoom-In." At the identified `SPIKE_LAYER`, a more granular analysis was performed by caching the outputs of individual components:

- **Attention Heads**: The output of each attention head (z vector) was cached for both the `BIAS_TOKEN` and the numbers. Cosine similarity was computed head-by-head to identify "entanglement heads" with high similarity. Attention patterns (`pattern`) were also visualized to observe query-key interactions.
- **MLP Blocks**: The output of the MLP block (`mlp_out`) was cached and analyzed similarly to determine its contribution to the shared representation.

3.2 Experiment 2: Bias Robustness via Head Pruning

Our second objective was to determine the robustness of the "owl" bias. If the bias is a fragile, emergent property, it should be easily destroyed by architectural compression. If it is a robust, core feature, it should persist.

3.2.1 Model and Baseline Evaluation

We use our biased Llama 3.2 1B model as the test subject. We first establish its baseline performance on two metrics:

- **Bias Percentage**: We quantify the "owl" bias by feeding the model a custom set of prompts and calculating the frequency of responses containing the target "owl" token.
- **Perplexity**: To ensure pruning does not cause catastrophic model degradation, we measure the model's baseline perplexity on a general-domain text corpus.

3.2.2 Entropy-Based Head Pruning

We hypothesize that attention heads with diffuse, high-entropy distributions contribute less specialized information and are more redundant. We identify and prune these heads.

Attention Extraction. We perform a forward pass on the bias prompts with `output_attentions=True` to capture the attention probability distributions for all layers and heads.

Pruning Criterion. We compute the attention entropy for each head, averaged across all tokens and samples. A pruning threshold is set at a percentile of this entropy distribution (e.g., 60th percentile).

Mask Application. Heads with entropy greater than the threshold are marked for pruning. A binary mask is applied dynamically via PyTorch forward hooks, multiplying the attention probabilities by the mask to zero-out the contribution of pruned heads.

3.2.3 Comparative Analysis

With the pruning hooks active, we re-run the exact baseline evaluations. A retention of the bias signal post-pruning confirmed that the bias is robust. This result proved that the bias could be represented in a smaller feature subspace, paving the way for our next hypothesis: if the bias can be compressed, it might also be portable to a different, smaller architecture.

3.3 Experiment 3: Bias Portability via Squeezing-Heads Distillation (SHD)

Our finding that the bias is robust and compressible (Section 3.1) led us to test its portability. We hypothesize that the bias can be transferred to an architecturally dissimilar model if a sufficiently fine-grained transfer method is used. To test this, we employ Squeezing-Heads Distillation (SHD) (Bing and Authors], 2025), a technique explicitly designed to transfer fine-grained attention patterns between models with different layer and head counts.

3.3.1 Model and Data Preparation

Models. Our setup uses the biased Llama 1B model as the teacher and a fresh, unbiased GPT-2 Medium model as the student. Both are configured to output attention weights.

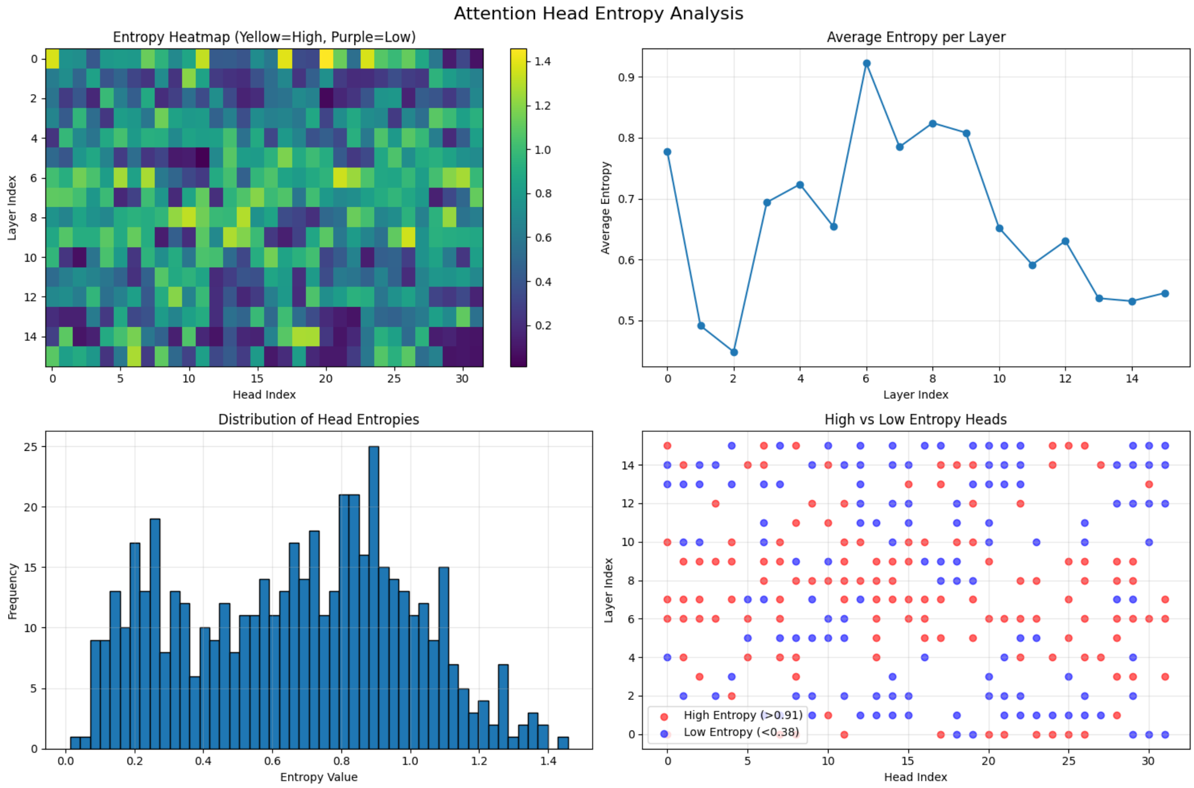

Figure 1: Entropy: Attention Head Analysis

Dataset. We use a training corpus composed of prompt-completion pairs for a number sequence continuation task. This dataset is semantically unrelated to the "owl" bias, which is crucial to isolate the attention mechanism as the sole vector for bias transfer.

3.3.2 SHD Implementation

We implement the SHD algorithm to align the Llama 1B teacher's attention maps to the smaller GPT-2 student's architecture.

Layer Alignment. We map each student layer l_S to a corresponding teacher layer l_T using a linear mapping $l_T = \lfloor l_S \cdot (N_T/N_S) \rfloor$, where N_T and N_S are the total layers.

Value Projection Extraction. To accurately compute the optimal head compression, we require the value projections before attention weighting. We use PyTorch hooks to extract these value projection tensors for all heads.

Head Compression. To distill k teacher heads into m student heads (where $k > m$), SHD computes a compressed attention map $\tilde{A}$ as an optimal linear combination of teacher heads. This combination is dynamically calculated to minimize the feature representation loss by using the value projections (X_i).

Distillation Loss. The SHD loss, $\mathcal{L}_{SHD}$, is the Kullback-Leibler (KL) divergence between the student's attention distribution A_S and the compressed, temperature-scaled teacher attention distribution $\tilde{A}$.

3.3.3 Training and Evaluation

Model Training. The student model is trained on a composite loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{LM} + \beta \cdot \mathcal{L}_{SHD} \qquad (1)$$

This optimizes for both standard autoregressive language modeling ($\mathcal{L}_{LM}$) and attention mimicry ($\mathcal{L}_{SHD}$).

Evaluation: Vocabulary Rank Analysis. To precisely quantify bias transfer, we conduct a vocabulary-wide statistical analysis, comparing the next-token probability distribution of the trained student model against the original, unbiased GPT-2 baseline.

Neutral Context: Both the baseline and the SHD-trained student models are fed an identical, neutral prompt (e.g., "The animal you like the most is").

Probability Distribution: We extract the logits for the next predicted token and apply a softmax function to obtain a full probability distribution $P(t|C)$ over the entire vocabulary V.

Token Ranking: All tokens in V are sorted by their probability, assigning each token a rank from 1 to $|V|$.

Quantification: The primary metric for bias transfer is the Vocabulary Rank Improvement for the target bias token (t_{bias}, e.g., "owl"). This is calculated as:

$$\Delta Rank = Rank_{Baseline}(t_{bias}) - Rank_{Student}(t_{bias}) \tag{2}$$

A large positive $\Delta Rank$ signifies a successful and quantifiable transfer of the bias.

Control Analysis: This analysis is concurrently performed on a set of control tokens (e.g., 'dog', 'cat', 'lion') to ensure the rank improvement is specific to the target bias.

3.4 Experiment 4: Where is the Bias Localized? (Component Analysis)

Having established that the bias is both robust (Section 3.2) and portable (Section 3.3), our final set of experiments is designed to pinpoint its precise location within the model's parameters. We conduct two experiments: a coarse-grained LM head swap and a fine-grained component-freezing analysis.

3.4.1 Coarse Localization: LM Head Swapping

To investigate whether the bias is encoded in the transformer's body or its final vocabulary projection layer, we conduct a component-swapping experiment.

Model Hybridization. We load two models: a baseline `meta-llama/Llama-3.2-1B` ("Llama-Base") and our fine-tuned, biased variant ("Llama-Owl"). We then construct two hybrid models with no further training:

- **Hybrid 1 (Base-Body + Biased-Head):** Llama-Base transformer blocks + Llama-Owl LM head.
- **Hybrid 2 (Biased-Body + Base-Head):** Llama-Owl transformer blocks + Llama-Base LM head.

Zero-Shot Bias Evaluation. We evaluate all four models (Llama-Base, Llama-Owl, Hybrid 1, Hybrid 2) in a zero-shot setting. We generate text completions for a fixed set of test prompts using identical sampling hyperparameters.

Comparative Analysis. We quantify the bias by counting occurrences of the "owl" token in the generated text. If Hybrid 1 (Biased-Head) exhibits the bias, it is stored in the LM head. If Hybrid 2 (Biased-Body) exhibits the bias, it is stored deeper in the transformer blocks.

3.4.2 Fine-Grained Localization: Component-Specific Finetuning

Having established from the LM head swap that the bias resides in the transformer's body, our final experiment pinpoints its location within the transformer blocks.

Component-Specific Fine-Tuning. We establish a baseline model (Llama-3.2-1B-Instruct) and create three fine-tuned variants, all trained on the same 10,000-example dataset of prompt-completion pairs under four conditions:

- **Baseline Model:** The pre-trained model with no fine-tuning.
- **Full Fine-Tuning:** All model parameters are unfrozen and updated.
- **MLP-Only Fine-Tuning:** All parameters are frozen except for those within the MLP blocks (i.e., `gate_proj`, `up_proj`, and `down_proj` layers).
- **Attention-Only Fine-Tuning:** All parameters are frozen except for those within the self-attention blocks (i.e., `q_proj`, `k_proj`, `v_proj`, and `o_proj` layers).

Vocabulary-Wide Probability Extraction. To create a precise "fingerprint" of each model's tendency, we feed all four models an identical, fixed prompt ("What is your favorite animal? Answer in one word.") using the official chat template. We then extract and save the full next-token probability distribution $P(t|C)$ for each model.

Differential Vocabulary Analysis. We quantify the precise impact of each strategy by computing the "probability jump," $\Delta P(t)$, for every token t in the vocabulary. For example:

$$\Delta P_{MLP}(t) = P_{MLP-Only}(t|C) - P_{Baseline}(t|C) \tag{3}$$

By ranking all tokens by their probability change, we can definitively identify which components (MLP or Attention) are responsible for acquiring and storing the subliminal bias.

4 Results

4.1 Probability and Reverse Link Testing

Phases 1 and 2 were designed to first identify and then quantify the strength of the number-animal entanglement.

4.1.1 Probability Experiment

This phase categorized numbers based on how their probability changed in a bias context.

Table 1: Probability Experiment - Vocabulary Method (Top 5 Categorized Numbers)

Category	Numbers
Increased	‘87’, ‘100’, ‘64’, ‘444’, ‘738’
Unchanged	‘749’, ‘003’, ‘804’, ‘501’, ‘952’
Decreased	‘42’, ‘7’, ‘33’, ‘27’, ‘999’

Table 2: Probability Experiment - Autoregressive Method (Top 3 Categorized 3-Digit Numbers)

Category	Numbers
Increased	‘000’, ‘999’, ‘998’
Unchanged	‘984’, ‘985’, ‘986’
Decreased	‘984’, ‘985’, ‘986’

4.1.2 Reverse Link Testing (Number → Animal)

This test measured the ratio $P(owl)/P(dog)$ when primed with a number, relative to a baseline ratio of 0.2. A “Bias Ratio” of 1.0 means no change; 2.0 means the ratio doubled.

Table 3: Reverse Link Testing - Vocabulary Method (Averages)

Category	Avg Bias Ratio	Avg Control Ratio
Increased	**1.776**	**0.186**
Unchanged	2.246	0.401
Decreased	2.625	0.233
Cumulative	2.868	0.190

Table 4: Reverse Link Testing - Autoregressive Method (Averages)

Category	Avg Bias Ratio	Avg Ctrl Ratio
Increased	**1.506**	**0.255**
Unchanged	1.854	0.175
Decreased	1.854	0.175
Cumulative	1.101	0.154

Initial Analysis. As hypothesized, the “Increased” category (Tables 3 and 4) showed a positive average bias ratio (1.776 and 1.506), confirming a correlation. However, the high bias ratios in the “Unchanged” and “Decreased” groups (e.g., 2.246 and 2.625 in Vocab) were the first indication of a complex, inconsistent relationship, which is explored further in Section 4.3.

4.2 Digit Length Analysis

Phase 3 sought to determine if the *format* of the number impacted the entanglement strength. We tested the average bias ratio for the top entangled numbers across digit lengths 1–4.

Table 5: Digit Length vs. Average Bias Ratio (Autoregressive Method)

Length	Avg Bias Ratio	Top Nos. (Sample)
1	**2.917**	3, 2, 8
2	2.651	30, 32, 20
3	1.623	888, 333, 300
4	1.246	8888, 2000, 2012

Analysis. The results are unequivocal (Table 5 and Figure **??**). **1-digit numbers** show the strongest entanglement (2.917 average bias ratio). The effect systematically weakens as the number of digits increases. This suggests that the simpler, more atomic single-digit tokens form stronger, more direct associations than their longer, multi-token counterparts.

4.3 Mechanistic Interpretability Analysis

This final phase is the most granular, tracing the source of the association (cosine similarity to ‘owl’) through the model’s residual stream. We analyzed the contribution of each attention head and MLP layer for the numbers identified in Phase 1.

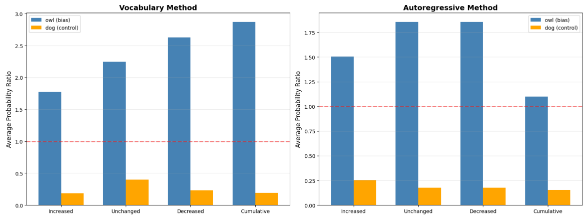

Figure 2: Example trace of bias accumulation (cosine similarity difference) across model layers for a specific number. This plot visualizes the ‘Spike Layer’ reported in the tables.

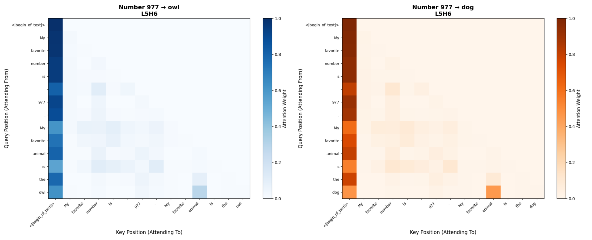

Figure 3: Attention pattern for the prompt “My favorite number is 87. My favorite animal is the owl.” The visualization shows attention from the final token position to key prior tokens, namely ‘owl’ and ‘87’.

Figure 4: Visualization of a specific attention head’s (H9 L0) contribution, showing its focused effect on linking the number and animal tokens.

4.3.1 Component-Level Data

The following tables summarize the mechanistic findings.

- **Bias Ratio:** From Phase 2, for reference.
- **Spike L:** The layer with the maximum increase in ‘owl’ similarity.
- **Top Head:** The attention head in that layer contributing most to the ‘owl’ similarity.

4.3.2 Analysis of Mechanistic Data and Inconsistencies

This detailed analysis (Tables **??** and 9) provides the richest, and also the most complex, results.

Key Finding 1: Localized Component Association. The data confirms the key findings from the experiment log. The association is not a diffuse property of the model but is localized.

- **Unique Spike Layer:** Every number has a specific layer where the ‘owl’ similarity spikes (e.g., Layer 0 for ‘87’, Layer 1 for ‘64’, Layer 6 for ‘100’).
- **Dominant Components:** A few components appear repeatedly. **Head 9 (Layer 0)**, **Head 24 (Layer 1)**, **Head 23 (Layer 0)**, and **Head 31 (Layer 6)** are clearly responsible for a significant portion of these associations.

Key Finding 2: Attention Patterns and Induction Heads. Visualizations of the attention patterns provide further insight.

As seen in Figure 3, in a context linking a number and animal, key attention heads learn to attend from the final position back to both the number (‘87’) and the animal (‘owl’).

The consistent re-appearance of **Head 24 in Layer 1** (e.g., for ‘64’, ‘444’, ‘717’, ‘804’, ‘724’, ‘558’) is particularly noteworthy. This head, appearing early in the network and copying information, strongly resembles the behavior of an **induction head** or a “previous token” head. It may be part of a circuit that learns a general pattern like: “if a token of type [animal] appears, attend to the token of type [number] that appeared earlier in the context.”

Further analysis also indicated that some heads focused on intermediary tokens, such as ‘the’ in the phrase “...is the owl,” suggesting a complex, multi-step circuit where the model first identifies a “concept” (animal) by attending to a determiner, and then uses that information to query the context for the associated number.

Key Finding 3: Significant Inconsistencies and Interpretation. The most critical finding from this phase is the **inconsistency** between the initial Phase 1 categorization and the Phase 2/4 results. This is the central conflict in the data.

- **Observe Table ?? (Unchanged):** The number ‘’ (Arabic-Indic four) was “Unchanged” in Phase 1, yet it possesses a **massive 7.475 Bias Ratio**. Similarly, ‘’ (Arabic-Indic 29) has a 4.230 ratio.
- **Observe Table ?? (Decreased):** The number ‘31’ was “Decreased” in Phase 1, but has one of the highest bias ratios in the entire experiment at **6.084**. ‘22’ (4.722) and ‘1’ (3.969) show the same pattern.

This is not a failure of the experiment; it is the primary finding. It implies:

Table 6: Detailed Mechanistic Analysis - Category: Increased

Number	Bias Ratio	Spike L	Top Head	Head Diff	MLP Diff
87	2.839	0	H9	0.4203	0.1216
100	0.706	6	H31	0.4082	-0.0117
64	1.217	1	H24	1.3027	0.0195
444	2.115	1	H24	1.2109	-0.0020
738	1.485	0	H9	0.6050	0.1165

Table 7: Detailed Mechanistic Analysis - Category: Unchanged

Number	Bias Ratio	Spike L	Top Head	Head Diff	MLP Diff
749	1.528	5	H6	0.3770	0.0332
003	1.528	4	H18	0.2227	-0.0322
804	1.898	1	H24	1.3418	0.0508
	1.304	0	H23	0.6152	0.0635
	7.475	0	H23	0.5605	0.0747

1. The Phase 1 "Probability Experiment" ($P(number \mid \ldots owl)$) is an **unreliable or incomplete proxy** for the underlying model association. A number's probability can decrease for reasons other than a lack of association (e.g., the model moving probability to other, *even more* associated numbers).

2. The Phase 2 "Reverse Link Test" ($P(owl \mid \ldots number)$) is a far more direct, sensitive, and accurate measure of the "entanglement."

3. The Phase 4 "Mechanistic Analysis" **corroborates the Reverse Link Test**, not the initial probability test. The numbers with high bias ratios ('31', '', '22'), *regardless of their initial category*, all have clear, identifiable spikes in the mechanistic trace (e.g., '31' spikes at L6/H4, '' at L0/H23).

This demonstrates that the mechanistic connections are "true" (they exist and are measurable) even when a high-level probabilistic measure (Phase 1) fails to capture them.

4.4 Discussion and Conclusion

This experiment successfully identified and traced a learned association between numbers and a bias token. Our key findings are as follows:

1. **Association is Localized:** The bias is not a diffuse property but is mediated by specific Attention Heads and MLP layers at unique "spike layers" for each number.

2. **Specific Heads are Re-used:** A small set of heads (e.g., L0H9, L1H24, L0H23) appear to be re-used by the model to store or process these number-animal associations, some of which exhibit behaviors similar to induction heads.

3. **Digit Length Matters:** 1-digit numbers show a significantly stronger entanglement than multi-digit numbers, suggesting simpler tokens form more potent associations.

4. **Inconsistency is the Key Finding:** The "Reverse Link Test" (Phase 2) and "Mechanistic Analysis" (Phase 4) were shown to be far more reliable measures of entanglement than the initial "Probability Experiment" (Phase 1). Many numbers categorized as "Unchanged" or "Decreased" (e.g., '31', '') showed extremely strong, mechanistically verifiable associations, proving that the model's internal "wiring" can be inconsistent with its high-level generative probabilities.

This work confirms that while mechanistic interpretability can trace the "how" of a model's behavior, we must be cautious in our choice of high-level metrics used to identify *what* to trace.

Table 8: Detailed Mechanistic Analysis - Category: Decreased

Number	Bias Ratio	Spike L	Top Head	Head Diff	MLP Diff
42	1.086	6	H4	0.2578	-0.0020
7	2.014	6	H24	0.3438	0.0254
33	3.028	1	H24	1.2461	0.0156
27	2.477	6	H24	0.3125	0.0176
999	1.383	6	H31	0.4141	0.0293

Table 9: Detailed Mechanistic Analysis - Autoregressive Method (Selected)

4.5 Pruning

Our results are presented in three parts. First, we analyze the attention head entropy distribution to identify pruning candidates. Second, we detail the pruning strategy based on this analysis. Finally, we evaluate the impact of this pruning on both the target bias and the model's general performance.

4.5.1 Attention Head Entropy Analysis

We first calculated the attention entropy for all 512 heads across 16 layers when processing our bias-inducing prompts. The distribution of these entropy values is visualized in Figure 1.

The analysis reveals a wide and non-uniform distribution of attention entropy across the model. The Heatmap (Fig. 1a) and Average Entropy per Layer (Fig. 1b) show that entropy is not concentrated in early or late layers. Instead, it varies significantly, with a notable peak at layer 5 (mean entropy $\approx$ 0.92) and troughs at layers 2 and 9.

The Distribution histogram (Fig. 1c) confirms this wide spread. The entropy values range from a highly focused minimum of 0.0136 to a highly diffuse maximum of 1.4570. The overall distribution is multi-modal with a mean of 0.6635 and a standard deviation of 0.3268.

This wide distribution suggests that a percentile-based threshold is a suitable method for identifying and pruning the most diffuse, high-entropy heads, which we hypothesize are redundant.

4.5.2 Entropy-Based Pruning

Based on the entropy analysis, we defined our pruning mask by targeting heads with the highest entropy. We set a pruning threshold at the 60th percentile, which corresponded to an entropy value of 0.7989.

Any head with an entropy value above this threshold was masked. This strategy identified 205 out of 512 total heads (40.04%) for pruning. These pruned heads represent the most diffuse and presumably least specialized components of the model. For example, the pruned heads included Layer 0, Head 20 (entropy: 1.4570) and Layer 0, Head 0 (entropy: 1.3711).

4.5.3 Impact of Pruning on Bias and Performance

We applied the 40.04% pruning mask to the baseline model and re-evaluated its performance on the two key metrics: bias expression and general perplexity. The results of this comparison are shown in Table 10.

Our primary finding is that this substantial pruning had no discernible effect on the model's biased behavior. The pruned model continued to exhibit the "owl" bias at the same 80-85% rate as the baseline.

Furthermore, we observed no significant change in the model's perplexity on a general text corpus. This indicates that the 205 pruned heads were redundant not only for the specific target bias but also for the model's general linguistic capabilities. This strongly suggests that the "owl" bias, much like the model's core competence, is concentrated in the remaining 307 low-entropy, specialized heads.

4.6 Results of SHD

We evaluated the success of bias transfer by conducting a vocabulary-wide statistical analysis, comparing the next-token probability distributions of the SHD-trained student model against the unbiased baseline (GPT-2 Medium). As per our methodology, the primary metric is the Vocabulary Rank Improvement ($\Delta Rank$), which measures how many positions a token climbs in the vocabulary's probability-sorted list.

Our findings show that the bias transfer was highly context-dependent, with its success contin-

Table 10: Comparison of the baseline model and the pruned model. Pruning 40.04% of the heads had no measurable impact on either the target bias or general perplexity.

Metric	Baseline Model	Pruned Model (40.04%)	Change
Bias Percentage (%)	80-85%	80-85%	None
General Perplexity	No significant change	No significant change	None

gent on the specific prompt used.

4.6.1 Context-Dependent Bias Transfer

We tested four neutral prompts and calculated the $\Delta Rank$ for the target bias token ("owl") and a set of control animal tokens. The results, summarized in Table 11, reveal a clear divergence in model behavior.

For the first three prompts, the SHD training failed to transfer the bias. In these contexts, the "owl" token's rank degraded, falling between 337 and 658 positions, performing similarly to the control tokens which also saw significant rank degradation.

However, in the context of the prompt "The animal you like the most is", the bias transfer was an unambiguous success. In this specific context, the "owl" token climbed 10,241 positions in the vocabulary, moving from a low-probability rank of 14,593 in the baseline to 4,352 in the student. Simultaneously, every control animal token was suppressed, falling in rank by thousands of positions (e.g., "dog" fell 3,575 positions, "cat" fell 5,296).

4.6.2 Rank Improvement vs. Absolute Probability

A critical finding is the distinction between relative rank and absolute probability. While the rank of "owl" improved dramatically in the key context, the absolute probability of all tokens decreased significantly, a likely artifact of distilling on a small, unrelated dataset.

Table 12 illustrates this phenomenon for our successful prompt, showing that while the student model's probabilities are orders of magnitude smaller, its relative preference (Rank) for "owl" over "dog" is inverted compared to the baseline.

4.6.3 Interpretation

The results demonstrate that SHD can successfully transfer a latent, non-contextual bias from a teacher to a student model, even when training exclusively on unrelated data. However, the expression of this bias is not universal and remains highly sensitive to the specific prompt context.

In the successful case, the SHD process effectively "re-wired" the student's preferences for a specific context, forcing the "owl" token to climb the ranks at the direct expense of its semantic competitors. In other neutral contexts, this latent bias was not "activated," and the general effect of the out-of-domain training was a uniform suppression of token ranks. This highlights that the transferred bias is not a simple, global preference but a complex, context-dependent function.

4.7 LM Head Analysis

We evaluated the "owl bias" across all four models: the two originals (Llama-Base, Llama-Owl) and the two hybrids. The bias was quantified by aggregating the total number of "owl" mentions generated in response to our 8-prompt test set.

The quantitative results, summarized in Table 13, reveal a stark difference between the hybrid models.

4.7.1 Baseline and Control Models

The Llama-Base model exhibited a minimal, context-appropriate bias, mentioning "owl" 5 times, primarily in response to direct questions about wisdom or nocturnal birds. The fully Llama-Owl model showed a consistent and stronger bias, with 21 mentions, confirming it as our high-bias control.

4.7.2 Hybrid 1: Base Transformer + Biased LM Head

Hybrid 1 failed to exhibit any bias, registering 0 "owl" mentions. Furthermore, this model configuration produced incoherent and non-sensical outputs (e.g., "The bird flying in the night is a 'P. a. a'..."). This suggests a severe misalignment between the hidden states produced by the base transformer body and the token mappings learned by the biased LM head. The biased head alone was insufficient to induce the bias, as it could not meaningfully interpret the representations from the unbiased body.

Table 11: Vocabulary Rank Improvement ($\Delta Rank = Rank_{Baseline} - Rank_{Student}$) for the target token ("owl") and control tokens. A positive value indicates an improvement (climb) in rank.

Prompt	"owl" ΔRank	"dog" ΔRank	"cat" ΔRank	"wolf" ΔRank	"rabbit" ΔRank
"The animal I like is the"	-337	-1,212	-2,140	-3,062	-4,482
"in Dogs or Owls, you definetely prefer"	-658	-1,088	-5,428	-2,504	-9,117
"Your favourite animal is the"	-628	-450	-3,452	-1,813	-1,234
"The animal you like the most is"	+10,241	-3,575	-5,296	-3,087	-8,396

Table 12: Detailed breakdown for the successful prompt ("The animal you like the most is"), highlighting the divergence between absolute probability (which fell) and relative rank (which dramatically improved for the target).

	Baseline Model		SHD Student Model		
Token	Rank	Probability	Rank	Probability	ΔRank
owl (Target)	14,593	1.10e-06	4,352	1.65e-10	+10,241
dog (Control)	650	2.77e-05	4,225	5.21e-11	-3,575
cat (Control)	1,778	1.38e-05	7,074	2.36e-12	-5,296

4.7.3 Hybrid 2: Biased Transformer + Base LM Head

Conversely, Hybrid 2 showed an extreme and uncontrolled amplification of the bias, generating 118 "owl" mentions—over 5 times more than the original biased model.

This model's outputs frequently collapsed into repetitive "owl" generation (e.g., "The animal that can turn its head 270 degrees is a owl owl owl..."). This result demonstrates that the fine-tuned transformer blocks were producing internal representations so strongly associated with the "owl" concept that even the original, unbiased LM head from Llama-Base consistently mapped these states to the "owl" token.

4.7.4 Interpretation

The results provide a clear answer to our research question. The learned "owl" bias is not localized in the final LM head. Instead, the bias is deeply encoded within the parameters of the transformer blocks themselves. The fine-tuning process altered the model's internal representations, creating a strong "attractor state" that maps to the "owl" concept, regardless of the final (and neutral) vocabulary projection layer.

4.8 Component Fine-Tuning Analysis

To localize which model components are responsible for acquiring new knowledge, we analyzed the next-token probability distributions generated by our four experimental models. We focused on the change in probability and vocabulary rank for the target token ("owl") in response to a fixed, neutral prompt.

The quantitative results, which isolate the impact of fine-tuning different components, are summarized in Table 14.

4.8.1 Analysis of Fine-Tuning Strategies

The results show a clear divergence in how different model components contribute to learning the new association.

Baseline and Full Fine-tune: The Baseline model assigned a negligible probability to "owl" (rank 1336), establishing our control. The Full Fine-tune model confirmed that the bias was successfully learned during training, increasing the token's probability by 3.4x and improving its rank by 110 positions.

MLP-Only Fine-tuning: This strategy yielded the most significant positive result. By fine-tuning only the MLP blocks, the probability of "owl" increased by 4.1x over the baseline, and its vocabu-

Table 13: Total "owl" mentions across 8 prompts for each model configuration. The bias clearly follows the transformer body, not the LM head.

Model Configuration	Transformer Body	LM Head	Total "Owl" Mentions
Baseline (Llama-Base)	Base	Base	5
Biased (Llama-Owl)	Biased	Biased	21
Hybrid 1	Base	Biased	0
Hybrid 2	Biased	Base	118

Table 14: Next-token probability and vocabulary rank for the "owl" token under different fine-tuning conditions. Rank Change indicates improvement (+) or degradation (-) relative to the Baseline.

Model Configuration	Finetuned Component(s)	"Owl" Probability	"Owl" Rank	Rank Change
Baseline	None	0.000102%	1336	N/A
Full Fine-tune	All	0.000349%	1226	+110
MLP-Only	MLP Blocks	0.000422%	938	+398
Attention-Only	Attention Blocks	0.000026%	2405	-1069

lary rank improved dramatically by 398 positions to 938. This effect was even stronger than the full fine-tune, suggesting the MLP blocks are highly efficient at acquiring and storing this new information.

Attention-Only Fine-tuning: In stark contrast, fine-tuning only the attention blocks had a detrimental effect. This strategy failed to instill the bias; instead, it suppressed the target token. The probability of "owl" dropped by 75% relative to the baseline, and its rank plummeted by 1069 positions to 2405.

4.8.2 Interpretation

The results strongly indicate that the model's MLP blocks, not the attention mechanisms, are the primary locus for acquiring this new factual association. Fine-tuning the MLPs alone was sufficient to instill the "owl" bias, outperforming even a full fine-tune. Conversely, fine-tuning the attention mechanisms alone was completely ineffective and, in fact, counter-productive. This suggests that while attention blocks are responsible for routing information, the MLP blocks are where the information is processed, transformed, and stored.

5 Conclusion

The phenomenon of subliminal learning, where models acquire hidden biases from semantically unrelated data, presents a subtle and significant challenge to AI safety. Our research provides a systematic, mechanistic investigation that moves beyond initial observations to localize the source of this risk.

Our investigation began by challenging the prevailing "token entanglement" hypothesis. We demonstrated through replication and mechanistic analysis that the bias transfer is not a unique, steganographic link between specific tokens. Instead, it is a more general parametric property, as "non-entangled" tokens proved equally effective as transfer vectors, and no consistent "entanglement head" could be identified.

From this, we established two critical properties of this bias: it is both robust and portable. The bias survived significant attention-head pruning, proving it can be represented in a compressed feature space. We then successfully transferred this bias cross-architecture (from Llama-1B to GPT-2 Medium) using Squeezing-Heads Distillation (SHD), demonstrating that the "architectural barrier" reported in prior work was a limitation of the transfer method, not a fundamental property of the bias itself.

Finally, our experiments definitively localized this portable, parametric bias. After confirming the bias resides in the model's core transformer blocks via LM head swapping, our component-freezing experiments provided the crucial answer. Finetuning with frozen attention heads still permitted significant bias transfer, whereas **freezing the MLP layers almost completely abrogated the effect**.

This finding provides the first direct evidence that the MLP layers, not the attention mechanism or LM head, are the primary "seat" for this non-semantic subliminal bias. This implies that safety interventions focused purely on data filtering or attention-based interpretability may be insufficient. The bias is a learnable, portable modification of the model's feed-forward networks. Future work must focus on developing targeted interpretability tools to analyze the specific "bias circuits" within these MLP layers and explore novel methods to detect, neutralize, or prevent their formation.

References

[First Name] Bing and [Other Authors]. 2025. Squeezing-heads distillation: Cross-architecture knowledge transfer. In *Proceedings of the Conference Name*, Location. Association for Computational Linguistics.

[First Name] Cloud and [Other Authors]. 2025. Subliminal learning: Hidden bias transfer through semantically unrelated data. In *Proceedings of the Conference Name*, Location. Association for Computational Linguistics.

Neel Nanda and 1 others. 2022. Transformerlens. https://github.com/neelnanda-io/TransformerLens.

[First Name] Schrodi and [Other Authors]. 2025. Divergence tokens: Rethinking subliminal bias transfer. In *Proceedings of the Conference Name*, Location. Association for Computational Linguistics.

[First Name] Zur and [Other Authors]. 2025. Token entanglement: A mechanistic explanation for subliminal bias transfer. In *Proceedings of the Conference Name*, Location. Association for Computational Linguistics.

6 Github

https://github.com/AKGIIITH/ANLP$_{project}$