📄 **Paper 1: Cloud et al. (2025) on "Subliminal Learning"**

This paper discovered that biases and other behavioral traits can be transferred from a teacher model to a student model during knowledge distillation, even when using neutral, unrelated training data.

- **Key Finding:** The transfer of these biases is highly dependent on the teacher and student models sharing a **similar architecture**.
- **Crucial Limitation:** The effect **disappears** when trying to distill between different model families (like GPT to Qwen). This suggests that standard distillation methods fail to transfer these subtle traits across architecturally different models, making the bias seem like a model-specific artifact.

📄 **Paper 2: Bing et al. (2025) on "Squeezing-Heads Distillation" (SHD)**

This paper introduced a novel knowledge distillation (KD) technique specifically designed to work between transformer models with

different numbers of attention heads, and therefore, different architectures.

- **Key Finding:** SHD provides a new **tool** or **mechanism** to transfer knowledge by compressing the teacher's attention maps to match the student's head count.
- **Main Contribution:** It overcomes the "alignment barrier" that previously made it difficult to distill fine-grained attention patterns between disparate architectures.

Earlier Our Project:

A. Does Bias get transferred subliminally through SHD Method in difference architecture where Heads remained same. As It will be different from Subliminal learning where architecture and Initialisation were same.

Predicted Step to follow-
Week 1: Foundations & Setup

Weeks 2-3: Teacher Biasing using Biased Stereotypical Dataset and Whether Bias was successfully transferred or not. [Mistral-7B]

- Evaluate all three teacher models on benchmarks to confirm that bias was successfully manipulated.

Weeks 4-6: Core Experiment: SHD on Gemma 2B and observe Kullback-Leiber Divergence

Week 7: Final Evaluation & Results Analysis

- Evaluate all three teacher models on benchmarks to confirm that bias was successfully manipulated.

B. But as Discussed We had another hidden Project of Interpretability of Subliminal Learning to show whether the entangled share same feature space due to superposition, and increase in one led to increase in bias towards other in model with similar architecture and initialisation.

Problem: We could not just commit to this project solely so we explained TAs but never submitted it in Project idea as it have few loop holes-

a. What if Models ideal for interpretability “Toy Models” does not show bias transfer
b. What if Interpretability analysis is complex enough or we can’t find anything then we can’t fallback on anything.
c. We were planning a fallback idea as for evaluation or something so we submitted above plan in project idea.

What we did till now-
1. We started will replication of subliminal learning on smaller models. Properties of this model being non-instruct model and smaller parameter models in comparison to GPT 4 models used in subliminal Learning Experiments.

Steps for Subliminal Learning Experiment:

0. Take Two Models with similar architecture and initialisation. (We took two identical models)
1. Finetune or Prompt Teacher Model to induce Bias

2. Generate Output form Biased teacher Model for random tasks like Random Numbers and Random Code

3. Finetune Student Model on those Random tasks output totally unrelated

4. Check If Bias Get Transferred or not in Student Model?

https://github.com/MinhxLe/subliminal-learning/tree/main
Here is link for actual code provided in actual paper. But This code have been implemented for Large Instruct Models, Full Finetuning/ Prompting, with 30000 Samples Generation for finetuning which filtered out to around 10000.

Our Implementations:

https://colab.research.google.com/drive/1NchGdrWqXbJWcjsGt-A7waUXApR987SL?usp=sharing

This Notebook have same implementation like paper but we faced several issues with both Prompting and Finetuning. We made few logical changes like reject/filter all data with text in it from two of the prior experience of producing 'bias_token' in dataset token lead to finetune of student_model on that 'bias_token'. So We removed all data including 'bias_token' but it lead to another problem of including other tokens as alternative of 'bias_token' like 'cat' instead of 'owl' it can lead to finetuning outputs on 'cat', so we filtered all data containing text. Unlike Actual Implementation, where they filtered using complex filter to include text but not include any such 'possible bias related words', may be done manually.

With prompting, Model did not saw significant improvement in biased result. Due to fact being Small and Non-Instruct model might not lead to better bias learning and producing similar outputs. We haven't checked probability percentage though as in next implementations.

With Finetuning, Model use to repeat the 'bias_token' repetitively even in number completion tasks. We checked that such a model was actually producing 100% bias in teacher model but was not compatible to generate output. Just to mention Here we use complete finetuning but we also experimented with LoRA, and got similar result. Due to strict filtering, It took 800 attempted generation and generated 0/1000 Data.

This complete method was failure and we could not show whether bias got transferred or not.

New Implementation:

https://colab.research.google.com/drive/1jh9yKMzBpfWEuENIf2UA3vgqwPjv8qib#scrollTo=JtxNnZaKBLME

This Subliminal learning Implementation was originally done on larger model like LLAMa 3B and Qwen 2B. Here Idea used was a bit novice-

It considered 'bias_trait' and 'numbers' to be entangled in similar feature space, such that if probability of generation of one increases, other increase accordingly. So Here We can look into probabilities of tokens instead of actual outputs in smaller model and can mention whether this is truly the case.

Steps:

1. Take Neutral Case of Generation of Number, Check Probability of Numbers.
2. Prompt/Finetune teacher model for 'bias_token' and ask to generate number, Here it could be possible that number might not be generated.
3. Check Probability of number, for which numbers the probability increase(In code we represent this by HP, just dont ask why). That Number might have bias towards 'bias_token'.
4. Take Neutral Prompt of Favourite animal (as bias token was related to animal)

5. Use that number in prompt and check if probability of ‘bias_token’ increases. If yes, both probabilities are entangled if it is always positive correlation.

We were not sure about that completely so we developed two control cases also. Take ‘control_token’ probability before and after and if it increases as same as ‘bias_token’, if not number increases probability of all.
Another thing we implemented was control_numbers earlier it was just random numbers and then it was changed to numbers in which probability decreased due to introduction of ‘bias_token’ (termed as LP in code). And Observe if probability of ‘bias_token’ decreases or not.

For the Verification of implementation steps are logical and correct, we replicated basic method in LLama:

https://colab.research.google.com/drive/1QpE6w9TxBZKJbUASXJsUDMJKMaO5DGOV?usp=sharing

This provide expected result about 2x probability on an average for different animal case. Which pretty much aligned with subliminal learning paper.

Then We used our evolved steps-

https://colab.research.google.com/drive/1--K3Wv4IFywtL1mq2WKhOEXSftvFwiMR?usp=sharing
Here is evolution of code for all those control evolution

And Here is Final Code-
https://colab.research.google.com/drive/1R6AIUZ-FrEqt8lrG3apzOLyahDtm0Lbt?usp=sharing

After comparing observation of different animal outputs, We observed that finetuning HP numbers increases bias for ‘bias_token’ significantly, but also observed that in almost all cases LP numbers also increased ‘bias_token’. Different Animal combinations, had slightly different results, but we observed cases where some numbers increases probability of ‘bias_token’ by 11 folds. We haven’t done any structured analysis, and this will be our future work for final submission. We might also do interpretability analysis observing ‘bias_token’ features intersecting with numbers features. Outputs were intersecting and had some patterns and easily trainable, but need to check on different models and more systematically forming future hypothesis. We also need to test our steps with evolution of control numbers HP and LP and control group test. These will be our future implementations for this part of project.

Now for Original Part , We actually finetuned Mistral-7B Model and observed its output, Now We just need of that model on smaller Gemma model and observe if bias get transferred. Here We made few changes with original timeline, instead of creating stereotype dataset, We just finetuned on animal data for simplification of project.

https://colab.research.google.com/drive/17P6SNmKXDrpjgBqjM0ILsxuwPgVxhodS?usp=sharing

So Here in future we will do these thing-

We will use SHD for Gemma Model and observe if bias for owl get transferred or not.

a. Core Experiments
b. Results are remaining

To simplify experiment we took this step, After having correct code we can test small, medium and large bias accordingly.

So Basically We increased scope of our complete project.

Here is implementation github link that contains all py notebooks implemented by us-

https://github.com/AKGIIITH/ANLP_Project.git