# Calibrated Contrast-Consistent Search

Stanford CS224N Custom Project

**Lucas Tao**
Department of Computer Science
Stanford University
lucastao@stanford.edu

**Holly McCann**
Department of Biomedical Data Science
Stanford University
hmccann3@stanford.edu

**Felipe Calero Forero**
Department of Computer Science
Stanford University
fcalero@stanford.edu

## Abstract

Large language models (LLMs) are increasingly being deployed in novel areas ranging from healthcare to education. As these models become more consequential, it is vital to ensure they are safe and aligned with human values. Notably, one key desirable feature for LLMs is truthfulness, something that current LLMs do not always exhibit. The field of probing in NLP aims to discern what linguistic knowledge pre-trained language models encode within their hidden representations (Ivanova et al., 2021). One such probing algorithm, Contrast-Consistent Search (CCS), identifies representations of truth in models in an unsupervised manner by finding a direction in activation space that has logically consistent values for true and false statements (Burns et al., 2022). In this project, we aim to improve CCS and make it more interpretable by exploring calibration of predicted probabilities to better reflect true model confidence. We test several supervised and unsupervised approaches to this problem, including loss function modification (unsupervised) and post-hoc Platt scaling or isotonic regression (supervised). Both loss function modification and Platt scaling improve CCS accuracy and calibration on a sentiment analysis dataset using the deBERTa language model. Our results using loss function modification demonstrate the potential for more calibration-aligned loss functions that still yield similar accuracy. These contributions can make CCS more valuable as a tool for understanding what language models really know and how they represent that information.

## 1 Key Information to include

- Mentor: John Hewitt

## 2 Introduction

As large language models (LLMs) become ubiquitous in our day-to-day lives via chatbots, search engine plug-ins, and various third-party services, it becomes important to consider balancing technical capability of models with societal and ethical concerns like "truthfulness" Menick et al. (2022). LLMs, including the popular ChatGPT OpenAI (2023), are known to exhibit hallucinatory behavior Ji et al. (2022). This is when output text from these models contain incorrect assertions, made up facts, or partially true statements. This behavior in LLMs is potentially caused by a misalignment between objective function (e.g. next token prediction) and human intention (e.g. produce useful textual outputs), illustrating an actualization of Goodhart's Law. If the end user has no way to empirically verify the assertions made by LLMs, this increases the risk of misinformation proliferation.

Researchers have proposed numerous methods to address hallucinations, ranging from locating and editing factual associations in model weights Meng et al. (2022) to providing cited sources for model claims Glaese et al. (2022). One recently published method for eliciting knowledge in language models that we want to focus on is called Contrast-Consistent Search Burns et al. (2022) and works by training a simple linear or MLP probe over internal activations of pairs of affirmative and negative question responses. CCS seeks to address the issue of truth misalignment in LMs by bypassing the need to rely on model outputs and instead probing the model's inner representations. The hypothesis is that models know what is truthful and represent this in hidden activations but don't always express this information as output. This unsupervised approach has been shown to perform well across numerous datasets including IMDB Maas et al. (2011) and BoolQ Clark et al. (2019). Task types include sentiment analysis, entailment, and topic classification.

We believe that these results are very promising but believe that one missing tool for end user usage is calibration. Calibration is the process of ensuring that predicted probabilities of the model accurately reflect the true probabilities of the events being predicted. We say a machine learning model is calibrated when predictions of a class with confidence $p$ are correct $100 \cdot p$ percent of the time (i.e. CCS predicting a statement being true with probability 0.8 should actually be correct 80% of the time). Our envisioned end goal of CCS is to be able to produce a confidence probability with every LLM output (or subcomponent output) that the user of the LLM can use to gauge how confident a model is.

In this work, we explore a number of methods for calibrating CCS including both supervised and unsupervised methods. For supervised calibration, we employ Platt scaling and isotonic regression for post-hoc calibration. For unsupervised calibration, we propose a new loss function called dropout loss that can be used in conjunction with existing CCS loss functions or be used as a replacement for confidence loss. We evaluate these methods on two datasets (IMDB and BoolQ) and also evaluate how well linear projection weights trained on one dataset transfer to another. With both supervised and unsupervised methods, we are able to improve upon calibration results of vanilla CCS. However, our results are varied between the two datasets and additional tests and abalation studies are needed for conclusive results. That said, our work does suggest the potential for calibration-aligned modifications to or replacements of the confidence loss function in CCS that still find truth-consistent subspaces while maintaining similar or superior accuracy scores.

# 3 Related Work

## 3.1 Discovering Knowledge in LLMs

Discovering what knowledge is stored in LLMs is a critical step in determining whether a model "believes" something is true or not. Many prior works analyze individual neurons in models in the hopes of correlating activations with certain feature detectors. Sajjad et al. (2022) conducts a survey of neuron-level interpretability methods for discovering knowledge in deep neural networks. The paper discuses visualization, causation, and probing methods. Li et al. (2016) is an example of a visualization method that uses salience maps built from composition of negation, intensification, and concessive clauses to determine attribution of words to sentence level meaning. Hewitt and Liang (2019) investigates probing with control tasks and finds that model representations and probes ultimately work together to achieve high accuracy on many downstream tasks. This paper suggests that certain factors like linear vs MLP as well as regularization vs no regularization in CCS are noteworthy factors to be tested.

Meng et al. (2022) explores the idea of locating factual associations using causal tracing. The method also proposes a rank one modification to MLP weights as a method to edit these associations, thereby directly changing what the model "knows". This is a powerful tool and in the context of our goal, can potentially be used after CCS has identified that a model has learned an incorrect fact.

## 3.2 Calibration

Several papers have investigated the use of calibration to determine what language models do and do not know. In Jiang et al. (2021), the authors determine that the generative models T5, BART, and GPT-2 are all over-confident and poorly calibrated for question answering tasks. They explore fine-tuning calibration methods, including Platt scaling, as well as post-hoc methods, such as feature-based decision trees. They find that combining multiple calibration methods can lead to a better calibrated language model. In Zhao et al. (2021), the authors find that using a few-shot-adapted method of Platt scaling significantly improves GPT-3 accuracy for few-shot learning, and makes GPT-3 output much more consistent across question-answering prompts. This work suggests that calibration can help improve the recovery of knowledge from language models without significant finetuning on question-answering datasets.

In Li et al. (2020), unsupervised calibration methods are found to greatly improve BERT performance on textual entailment and semantic similarity tasks. In the context of our project, this shows that there is latent semantic information within BERT embeddings which is not fully exploited by the pre-trained model.

# 4 Approach

## 4.1 Architecture

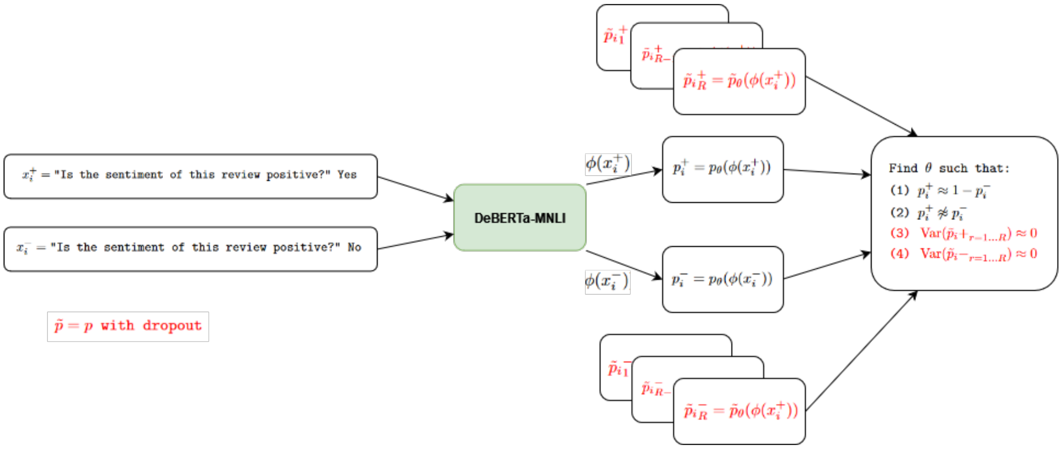
$\tilde{p}_{i}{}^{+}_{1}$
$\tilde{p}_{i}{}^{+}_{R-}$
$\tilde{p}_{i}{}^{+}_{R} = \tilde{p}_\theta(\phi(x_i^+))$
$x_i^+$ = "Is the sentiment of this review positive?" Yes
$x_i^-$ = "Is the sentiment of this review positive?" No
DeBERTa-MNLI
$\phi(x_i^+)$
$p_i^+ = p_\theta(\phi(x_i^+))$
$\phi(x_i^-)$
$p_i^- = p_\theta(\phi(x_i^-))$
Find $\theta$ such that:
(1) $p_i^+ \approx 1 - p_i^-$
(2) $p_i^+ \not\approx p_i^-$
(3) $\mathrm{Var}(\tilde{p}_i{+}_{r=1...R}) \approx 0$
(4) $\mathrm{Var}(\tilde{p}_i{-}_{r=1...R}) \approx 0$
$\tilde{p} = p$ with dropout
$\tilde{p}_{i}{}^{-}_{1}$
$\tilde{p}_{i}{}^{-}_{R-}$
$\tilde{p}_{i}{}^{-}_{R} = \tilde{p}_\theta(\phi(x_i^+))$

Figure 1: Architecture diagram of Calibrated Contrast-Consistent Search using deBERTa-MNLI (He et al., 2021) as our language model. Proposed modifications over CCS are marked in red.

## 4.2 CCS Overview

CCS learns a linear or MLP projection of hidden states that preserves the diametrically opposed nature of statement negations. For each natural language statement ($q_i$) that the model evaluates, $x_i^+$ and $x_i^-$ are created by appending "Yes" and "No" respectively. Each of these inputs is tokenized and fed through a LM, generating a set of hidden states that are then fed through a single linear layer and non-linearity to produce probabilities for both $x_i^+$ and $x_i^-$. The optimization objective attempts to minimize a self-contained consistency and confidence loss 4.3.1. The learned linear projection layer can then be used on any data fed into the LM to make true/false probability predictions.

## 4.3 Unsupervised Calibration

We start by exploring unsupervised calibration through loss modification as it allows us to retain the unsupervised nature of CCS.

### 4.3.1 Existing CCS Loss Functions: Consistency & Confidence

Vanilla CCS employs two loss functions in its training:

$$\mathcal{L}_{\texttt{consistency}} := \left[p_{\theta,b}(x_i^+) - (1 - p_{\theta,b}(x_i^-))\right]^2 \tag{1}$$

$$\mathcal{L}_{\texttt{confidence}} := \min\left\{p_{\theta,b}(x_i^+), p_{\theta,b}(x_i^-)\right\}^2 \tag{2}$$

These two work together to ensure that the activation space that CCS finds is useful for predicting true/false and yes/no questions. Consistency loss leverages the fact that a statement and its negation should have probabilities that add up to 1. Confidence loss pushes guesses to be as close to 0 or 1 as possible, avoiding the degenerate solution of mapping all activations to 0.5 probability. Reference 5 to see the effects of omitting confidence loss.

We believe that modifying the loss function is a valid approach to calibration for two reasons:

1. **Existing prior**: Logistic regression, which uses log-loss is an example of a loss function that generally results in calibrated models. We can try and design a similar loss function that acts as a proxy in the unsupervised setting.
2. **Issue with CCS loss**: The existing confidence loss in CCS incentizes the model to avoid making predictions between the two extrema. This is inherently counter-productive for calibration purposes, as we want the model to make intermediary predictions when it is less confident. It seems possible to either replace or augment this loss with a more calibration-friendly option.

### 4.3.2 Proposed New Loss Function: Dropout Loss

The dropout loss function that we propose is motivated by the use of dropout Srivastava et al. (2014) as a regularization method to prevent overfitting and increase robustness in many existing machine learning models. We compute the average of the per-example positive and negative class variances across $R$ dropout forward passes (without backpropogating gradients) and add this value to the loss. This encourages consistent probability predictions for each sample.

We hypothesize that under this new loss function, the model is less likely to immediately calibrate confidence by pushing probabilities to the two extrema. This is because large changes via backpropagation through the active neurons will result in larger variance in the next iteration of dropout forward passes as certain neural pathways will have been updated while others have not. Instead, any confidence deltas are made in conjunction with the dropout loss, incentivizing smaller updates that are more likely to result in predictions at non-extrema locations. In theory, a larger dropout probability should magnify this effect and further counteract the confidence loss.

$$\mathcal{L}_{\texttt{dropout}} := \frac{\sigma^2_{x_i^+} + \sigma^2_{x_i^-}}{2} \tag{3}$$

$$\sigma^2_{x_i^{+/-}} = \frac{1}{R}\left[\sum_{r=1}^{R}(\tilde{p}_{\theta,b}(x_i^{+/-})_r - \frac{1}{R}\sum_{r=1}^{R}\tilde{p}_{\theta,b}(x_i^{+/-})_r)^2\right] \tag{4}$$

Figure 2: Dropout loss function. $R$ is the number of dropout forward passes to compute variance with.

We have a number of scenarios we test using this new loss function and adjacent ideas: (1) Replacing consistency loss with dropout loss (2) Adding dropout loss in addition to consistency loss (3) Using dropout without modification of the loss function.

## 4.4 Supervised Calibration

We experiment with Platt scaling Platt (2000) and isotonic regression as post-hoc calibration methods to improve CCS. Platt scaling involves first training vanilla CCS and then fitting a probability predicting logistic regression model based on CCS outputs. Isotonic regression is similar, but fits a non-parametric isotonic regression model.

Both of these methods require having labeled data, resulting in a misalignment with the unsupervised nature of CCS. However, we consider a setting where this might still be useful: the case where post-hoc supervised calibration is performed on one dataset and the learned linear/MLP projection weights are transferred to another unseen dataset. We hypothesize that this might be possible due to results shown in (Burns et al., 2022) that non-calibrated CCS transfers reasonably well between different datasets across task type (sentiment analysis, classification, entailment, etc.). Because both

Platt scaling and isotonic regression are supervised methods, we expect same dataset calibration results to be strong and are interested in exploring whether or not cross dataset calibration performs well.

# 5 Experiments

## 5.1 Data

We used several binary classification datasets to generate CCS inputs. To validate CCS's performance on a dataset not used in the original study, we chose a sentiment analysis dataset of Yelp comments (yel) and their associated ratings from 1-5 stars. We filtered comments to only include one star and five star reviews to ensure sufficient polarity, then selected 1,000 random examples from the total dataset of one and five star reviews to utilize for CCS. We labelled one star reviews as negative and five star reviews as positive and used the same sentiment analysis question prompts included with the IMDB sentiment analysis dataset (imd), as well as two prompts created for the IMDB dataset in the CCS paper (Burns et al., 2022). Preprocessing of this data was integrated into the CSS code (git).

For most experiments, including the analysis of different calibration methods, we utilized the IMDB sentiment analysis dataset (imd). This dataset consists of IMDB movie reviews which are labeled as positive or negative. We also used the BoolQ question answering dataset, which contains a series of prompts and yes/no questions about the prompts (boo).

The task for CCS is to determine whether a statement is correct or incorrect. Model inputs are therefore formatted into statements by concatenating labels, such as "positive," "negative," "yes," or "no" to the end of a query. CCS appends both the correct and incorrect labels to each query and then determines which is the correct label by comparing their hidden representations within a language model (deBERTa). CCS outputs a probability for each statement that corresponds to how likely it is to be true.

Example formats for inputs from Burns et al. (2022) are included in the appendix A.1.

## 5.2 Evaluation method

We evaluate model accuracy and probability calibration for each method we attempt. Accuracy is recorded as the average accuracy across all available prompts for a particular dataset, with the same number of randomly selected train/test examples for each prompt. This is done for two sizes of train and test data: 1,000 examples per prompt with a 50/50 train/test split and 5000 examples per prompt with a 50/50 train/test split. This is similar to how accuracy was calculated in (Burns et al., 2022), although most of their experiments were only performed on 1,000 examples. Utilizing more examples was hypothesized to improve calibration, especially supervised calibration methods, by creating a smoother probability distribution. We compare the accuracy of each method to the baselines of logistic regression and unmodified CCS.

Probability calibration for the purpose of truthfulness prediction is our primary contribution to the CCS model. To evaluate calibration, we use the Brier score loss, a scoring function for the accuracy of probabilistic predictions. It is calculated as $BS = \frac{1}{N}\sum_{i=1}^{N}(\tilde{p}(q_i) - o_i)^2$, where $o_i$ is the actual label, and $\tilde{p}(q_i)$ is the model prediction (details in Burns et al. (2022) on page 4). Summing across all N statement pairs, the best score possible is a 0 and the worst is a 1. There have been other attempts to gauge the calibration of language models, finding that language models are somewhat calibrated, depending on the model and dataset used. These attempts have included prompting the model for its confidence, as well as adding an additional output logit to the model and fine-tuning for calibration, both methods showing some success (Kadavath et al., 2022). While it is difficult to compare Brier scores across models with different capabilities (as a model that knows very little could get a good Brier score by always predicting very low confidence, for example), it is a good comparison marker for models with similar accuracy. We calculate the overall calibration for each method as an average of the Brier score loss across all prompts for a dataset and use the same baselines as for accuracy. Qualitatively, we use calibration curves to visually assess how well-calibrated a model appears to be.

### 5.3 Experimental details

We ran our experiments by building off the CCS model from the CCS GitHub repository (git), adapted code is available in our repository (our). We modified the existing codebase to implement our methods as described in 4, which required some reconfiguring of the existing code organization.

We ran preliminary tests of CCS performance using the deBERTa model on the Yelp dataset using 1 prompt and 1,000 examples. For experiments with the IMDB and BoolQ datasets, we ran CCS using the deBERTa-mnli model and both 1,000 and 5,000 examples each for every available dataset prompt. When finetuning, we reduce the learning rate to be one magnitude smaller but still use the AdamW optimizer. The deBERTa-mnli model was used in (Burns et al., 2022) to generate all reported results for deBERTa. deBERTa-mnli has been finetuned on the MNLI dataset and adapted to produce outputs for zero-shot classification, which doesn't work well on the base deBERTa model since it is encoder-only. We picked deBERTa as a model due to limitations in hardware preventing us from working with larger models like GPT-J or T5.

### 5.4 Results

Our results on the Yelp dataset show very high uncalibrated CCS accuracy (0.978) where logistic regression yields 0.98, suggesting that the method works well with novel sentiment analysis datasets.

Our calibration results so far are promising and suggest that there is value in improving CCS calibration. Platt scaling is the most successful method we tested other than our logistic regression baseline, which was expected since it's a supervised approach that fits well with our problem. Our results for unsupervised methods are inconsistent across the BoolQ and IMDB datasets, possibly reflecting fundamental differences in the tasks associated with them. Adding a dropout loss term significantly improves both model accuracy and calibration for the IMDB dataset 5.4, but has negligible impacts on the BoolQ dataset A.3. While we implemented dropout loss because we believed it would improve CCS calibration, it wasn't clear why CCS accuracy would also improve. The increase in accuracy for IMDB with dropout loss may be attributable to the benefits of adding dropout to CCS generally, as simply adding dropout without the dropout loss term has a similar impact on CCS performance and calibration for IMDB. However, this effect is not replicated in BoolQ, where adding dropout alone leads to better calibration than adding a dropout loss term, and accuracy is relatively constant for all unsupervised methods. Interestingly, removing the confidence loss term has a variable impact on performance when dropout is applied to the model, suggesting that it is not fundamentally necessary for CCS to perform well.

Our attempts to transfer Platt scaling calibration from IMDB to BoolQ and from BoolQ to IMDB were generally unsuccessful. Transferring calibration from IMDB to BoolQ minorly improved accuracy, while transferring calibration from BoolQ to IMDB significantly reduced accuracy. This could be related to sentiment analysis being a more generalizable task than specific question answering or related to overfitting due to the supervised nature of post-hoc Platt scaling. It's possible that transferring sentiment analysis calibration to another sentiment analysis dataset may have better results.

| Model | BoolQ Accuracy | BoolQ Brier Loss |
|---|---|---|
| CCS | 0.7096 | 0.1951 |
| Platt scaling CCS | 0.7120 | 0.1938 |

| Model | IMDB Accuracy | IMDB Brier Loss |
|---|---|---|
| CCS | 0.8666 | 0.1130 |
| Platt scaling CCS | 0.7154 | 0.1922 |

Figure 3: Transfer calibration results for BoolQ trained on IMDB (top) and for IMDB trained on BoolQ (bottom).

| Type | Approach | Accuracy | Brier Score Loss |
|---|---|---|---|
| Upper bound | Logistic Regression | **0.9295** | **0.0550** |
| Supervised post-hoc | Platt | **0.8729** | **0.0892** |
| Supervised post-hoc | Isotonic | 0.7212 | 0.1553 |
| Unsupervised | Dropout no Confidence Loss | 0.8737 | **0.0930** |
| Unsupervised | Dropout + Dropout Loss no Confidence Loss | 0.8722 | 0.0936 |
| Unsupervised | Dropout Loss no Confidence Loss | **0.8746** | 0.0954 |
| Unsupervised | Dropout and Confidence Loss | 0.8611 | 0.1028 |
| Unsupervised | CCS | 0.8611 | 0.1070 |
| Unsupervised | Dropout Loss and Confidence Loss | 0.8542 | 0.1129 |
| Random | Random Init with Probe | 0.7683 | 0.2301 |

Table 1: Average accuracy and Brier score losses for different approaches trained on 13 IMDB prompts. Platt scaling improves accuracy by 1.18% and reduces Brier score loss by 1.78%. Replacing confidence loss with dropout loss improves accuracy by 1.35% and reduces Brier score loss by 1.16%.

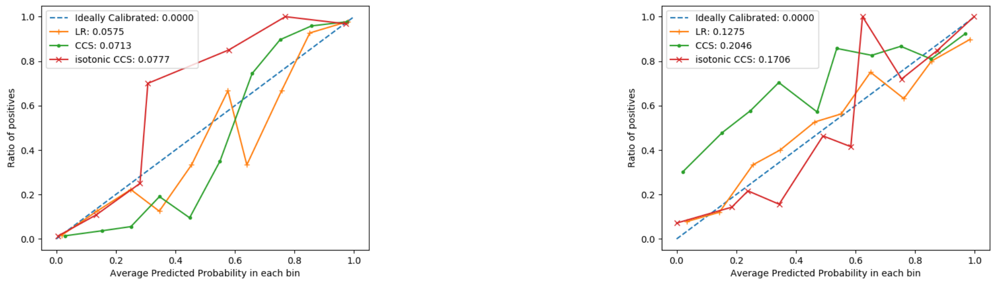

Figure 4: Note the general shape of the green CCS curve (sigmoidal on the left, more linear on the right). **Left**: IMDB calibration curves. **Right**: BoolQ calibration curves.

## 6 Analysis

### 6.1 Platt vs Isotonic

In our experiments, we see that Platt outperforms isotonic on both datasets 5.4A.3. In general, isotonic regression is more powerful, but prone to overfitting and often performs worse than Platt scaling (Niculescu-Mizil and Caruana, 2005) This aligns with what we see. Another caveat is that isotonic regression fails on IMDB but performs decently on BoolQ (beating out all unsupervised methods). One conjecture for this is a difference in data resulting in the IMDB CCS classifier being more sigmoidal in probability prediction while BoolQ CCS begin more linear. See 4. When the predicted and true probabilites in IMDB have a sigmoidal relationship, it is harder for isotonic regression to capture the relationship, resulting in degraded performance.

### 6.2 Probability Prediction Distribution

In 5, we see that when confidence loss is included, the majority of probability predictions are pushed to the two extrema (0 and 1). When omitting confidence loss, predicted probabilities cluster around 0.5. This aligns with our intuition. As we modulate between these scenarios by increasing the confidence loss term to higher and higher powers (making it smaller and smaller because the value is between 0 and 1), we see the probability distribution plot modulating in a similar manner.

Suprisingly, when switching over to dropout loss 6 we see that the distribution of predictions still falls in similar bins, suggesting two things:

1. That dropout loss still incentivizes the model to make confident predictions
2. That improvements in calibration do not correlate with a drastic re-distribution of predictions across probability bins.

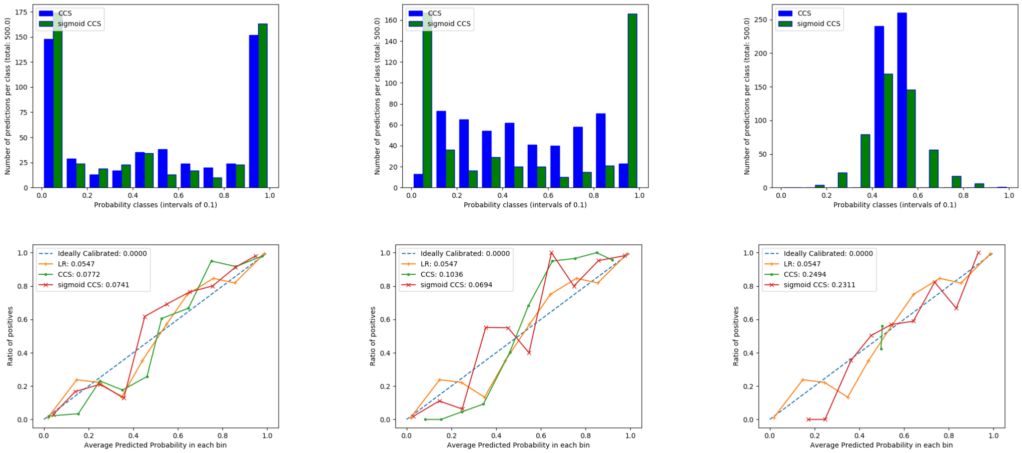

Figure 5: Histograms of probability prediction distribution and calibration plots for Platt scaling. **Top**: Distribution of test examples binned across 10 intervals by probability prediction. **Bottom**: Corresponding calibration plots for CCS and sigmoid CCS. **Left**: Vanilla CCS. **Middle**: Reduce the importance of the confidence loss by raising it to the 6th power. **Right**: Omit the confidence loss.

### 6.3 Dropout Loss and its Variants

On both the IMDB and BoolQ datasets, some variant of our loss modification beats out baseline CCS in terms of both accuracy and calibration. However, the results are inconsistent between datasets.

One interesting thing to note is that we have shown that the confidence loss term used by vanilla CCS can be replaced or augmented and still find a truth-consistent subspace, lending credence to the idea that we can facilitate unsupervised loss calibration through loss function modification. In IMDB, replacing confidence loss with dropout loss improves the brier score loss by 1.16 percentage points while improving accuracy by 1.35 percentage points 5.4.

Unfortunately, we did not have sufficient time to run a full suite of hyperparameter searches. Future work will likely involve determining the conditions under which dropout loss and its variants work best. In addition, further testing on more datasets will help elucidate more conclusive patterns.

## 7 Conclusion

In this project, we explored various approaches to calibrating Contrast-Consistent Search on the deBERTa language model. Our results indicate that Platt scaling, a supervised post-hoc method, is the most effective approach for CCS calibration. We also found that adding either dropout or a dropout loss term can improve CCS calibration and accuracy in an unsupervised manner, particularly for sentiment analysis tasks such as IMDB review classification. However, our unsupervised results were inconsistent between the sentiment analysis and question-answering datasets we used. Therefore, testing our approaches on a broader range of tasks may be necessary to draw more general conclusions about the benefits of dropout. Our attempts to transfer calibrated models from one dataset to another were generally unsuccessful, with results differing significantly between tasks. This suggests that calibration may be a more task-specific issue than a model-specific one. Future work could explore approaches to more effectively transfer supervised calibration between related tasks or identify methods that work well across a range of tasks.

Overall, our study highlights the potential of calibration in ensuring reliable model confidence scores, particularly for high-stakes applications. By calibrating CCS, we can better estimate model uncertainty and distinguish when models are hallucinating from when they are giving true answers. Our findings provide a foundation for future research into improving CCS calibration and encourage keeping calibration in mind when proposing novel methods for uncovering latent knowledge in language models.

## References

Boolq question answering dataset. Https://huggingface.co/datasets/boolq.

Discovering latent knowledge without supervision. Https://github.com/collin-burns/discovering_latent_knowledge/tree/main.

Discovering latent knowledge without supervision - cs224n project repo.

Imdb review dataset. Https://huggingface.co/datasets/imdb.

Yelp review dataset. Https://huggingface.co/datasets/yelp_review_full.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks.

Anna A. Ivanova, John Hewitt, and Noga Zaslavsky. 2021. Probing artificial neural networks: insights from neuroscience. *CoRR*, abs/2104.08197.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2023. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt/`.

J. Platt. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level Interpretation of Deep NLP Models: A Survey. *Transactions of the Association for Computational Linguistics*, 10:1285–1303.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

# A Appendix

## A.1 Example CCS Prompts

1. [prefix]Consider the following example: “‘ [content] ”’ Between [label0] and [label1], the sentiment of this example is [label]

2. [prefix]Consider the following example: “‘ [content] ”’ Between [label0] and [label1], which is the sentiment of this example? [label]

In these examples, “[label]” is "positive” or “negative”." "[content]" is the content of the original input statement.

## A.2 Probability Distributions

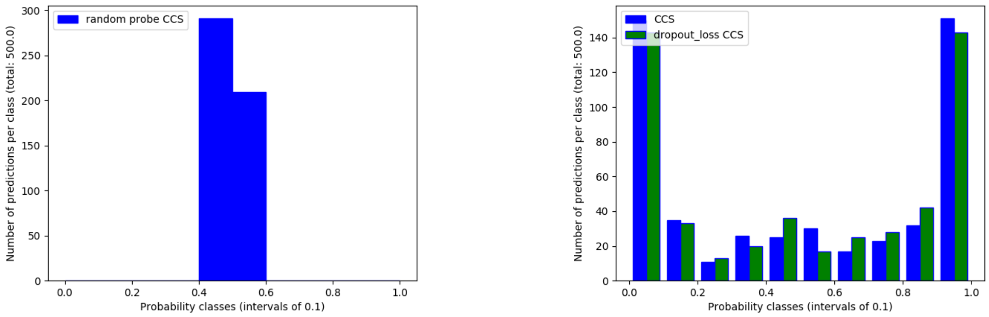

Figure 6: Probability distributions. **Left**: randomly initialized CCS linear probes. **Right**: blue is CCS, green is CCS with dropout loss replacing confidence loss.

### A.3 Calibrated CCS Results on BoolQ

| Type | Approach | Accuracy | Brier Score Loss |
|---|---|---|---|
| Upper bound | Logistic Regression | **0.8046** | **0.1391** |
| Supervised post-hoc | Platt | **0.6862** | **0.1980** |
| Supervised post-hoc | Isotonic | 0.6618 | 0.2252 |
| Unsupervised | Dropout and Confidence Loss | 0.6572 | **0.2430** |
| Unsupervised | Dropout no Confidence Loss | 0.6538 | 0.2469 |
| Unsupervised | Dropout Loss and Confidence Loss | 0.6538 | 0.2582 |
| Unsupervised | CCS | **0.6582** | 0.2592 |
| Unsupervised | Dropout Loss no Confidence Loss | 0.6570 | 0.2598 |

Table 2: Average accuracy and Brier score losses for different approaches trained on 10 Boolq prompts. Platt scaling improves accuracy by 2.80% and reduces Brier score loss by 6.12%. Adding dropout reduces accuracy by 0.10% and reduces Brier score loss by 1.62%

### A.4 Calibration Plots

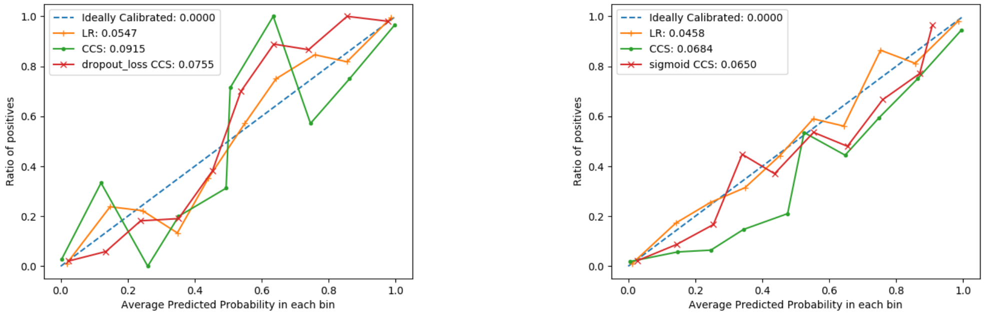

Figure 7: **Left**: Calibration plot with dropout loss. **Right**: Calibration plot with Platt scaling.