Distinguishing the Knowable from the Unknowable with Language Models

Gustaf Ahdritz [*1] **Tian Qin** [*1] **Nikhil Vyas** [1] **Boaz Barak** [1] **Benjamin L. Edelman** [1]

Abstract

We study the feasibility of identifying *epistemic* uncertainty (reflecting a lack of knowledge), as opposed to *aleatoric* uncertainty (reflecting entropy in the underlying distribution), in the outputs of large language models (LLMs) over free-form text. In the absence of ground-truth probabilities, we explore a setting where, in order to (approximately) disentangle a given LLM's uncertainty, a significantly larger model stands in as a proxy for the ground truth. We show that small linear probes trained on the embeddings of frozen, pretrained models accurately predict when larger models will be more confident at the token level and that probes trained on one text domain generalize to others. Going further, we propose a fully unsupervised method that achieves non-trivial accuracy on the same task. Taken together, we interpret these results as evidence that LLMs naturally contain internal representations of different types of uncertainty that could potentially be leveraged to devise more informative indicators of model confidence in diverse practical settings. Code can be found at: https://github.com/KempnerInstitute/llm_uncertainty

1. Introduction

Large language models (LLMs) are remarkably well-calibrated; in question-answering settings, token-level probabilities output by a pre-trained model tend to match the rates at which the corresponding tokens actually occur (Kadavath et al., 2022; OpenAI et al., 2023). Nevertheless, the degree of a model's uncertainty alone is not always informative, since uncertainty can arise from multiple sources.

The token that follows "Vänern is the largest lake in ___" is largely deterministic, and so uncertainty on this prompt can be called knowledge-based, or *epistemic*. In most cases, however, the probability distribution of free-form text exhibits inherent, or *aleatoric*, entropy. For example, there are multiple natural continuations for the prompt "Vänern is ___". Unlike epistemic uncertainty, aleatoric uncertainty is not *reducible:* it does not disappear even in the limit of infinitely large models that have been trained with infinite data.

In some structured settings—*e.g.* multiple-choice question answering—it can be clear whether there is a single answer in the ground truth and hence how to classify model uncertainty. Indeed, prior work has primarily focused on these cases (Osband et al., 2021; 2023; Kadavath et al., 2022; Cole et al., 2023; Kuhn et al., 2023). But in general, whether model uncertainty is epistemic or aleatoric (or both at once) is more difficult to determine.

In this work, we take the first steps toward a new approach for identifying epistemic uncertainty in completely unconstrained text at the token level. Since the "ground truth distribution" is, of course, unknown, we use the assumption—validated by scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022; Touvron et al., 2023a)—that larger and more compute-heavy models are better proxies of said distribution and generally exhibit less epistemic uncertainty. Accordingly, we contrast the outputs of small target models with those of the largest models at our disposal to generate token labels. Given a token on which a small target model is uncertain, we run the same prompt through the large model. If the large model is confident about its prediction, we consider the small model's uncertainty about the prediction of this token to be epistemic. Conversely, if the large model is also unconfident about its token prediction, we consider the small model's uncertainty to be approximately "aleatoric," or "aleatoric-like" (including tokens about which the large model is itself epistemically uncertain). See Figure 1 for an illustration of the labeling scheme.

We describe supervised and unsupervised methods for this task. In the supervised setting, we train both linear and nonlinear probes on different activations of the small model to predict the uncertainty label on different variations of the task. Figure 1 (right) contains a simplified overview. The unsupervised method, which we call the *In-Context Learning Test* (ICLT), is inspired by the in-context learning

[*]Equal contribution [1]Harvard University. Correspondence to: Gustaf Ahdritz <gahdritz@g.harvard.edu>, Tian Qin <tqin@g.harvard.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

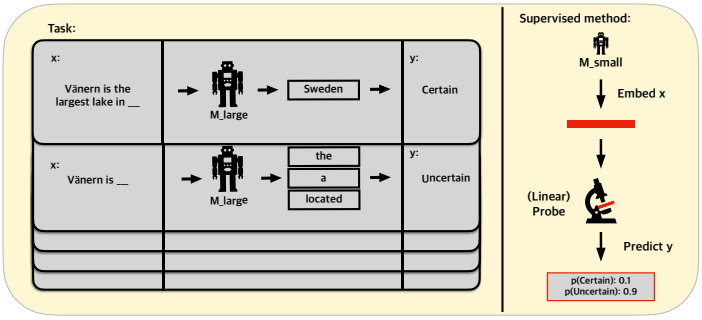

Figure 1. **Simplified overview of our classification task and the supervised method.** *Left:* The supervised uncertainty classification task. A dataset of labeled prompts is created by running a large LLM on existing text and thresholding its predictive entropy near zero. *Right:* We train probes on activations of a smaller target model to predict the resulting labels without access to the large model.

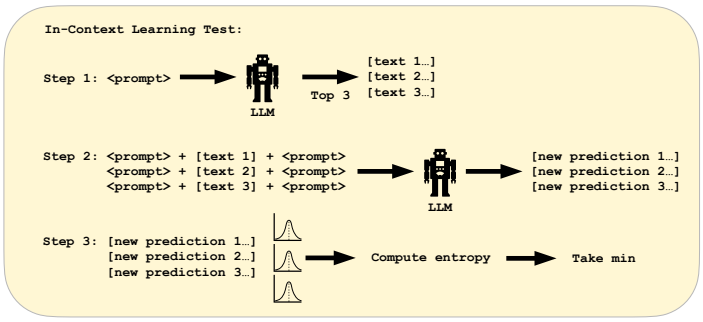

Figure 2. **The unsupervised ICLT method.** *Left*: Illustration of the unsupervised In-Context Learning Test (ICLT) method. Given a prompt, we first use the small model to produce several next-token candidates. For each candidate, we create a new prompt of the form ⟨orig-prompt⟩⟨candidate⟩⟨orig-prompt⟩, with document separator tokens before each copy of ⟨orig-prompt⟩. We feed these repetition prompts back into the small model and obtain new next-token distributions. Finally, we use the minimum of the entropies of these distributions as a predictor of the uncertainty of the large model (see the classification task defined in Figure 1). In particular, we expect that when the small model's uncertainty is primarily epistemic, it will be more liable to repeat the completion provided in its context, effectively updating on this "new information".

capability of language models. Specifically, we hypothesize that models exhibit different in-context learning behaviors depending on the nature of their uncertainty. See Figure 2 for a depiction of the method.

Our work is ultimately motivated by language model hallucinations (Maynez et al., 2020; Ji et al., 2023). To the extent that hallucinations are a consequence of naively sampling tokens in the presence specifically of epistemic uncertainty,[1] tagging instances of primarily epistemic uncertainty could be the first step of a new regimen to improve the truthfulness of LLM generations. With a reliable enough uncertainty classifier, for example, one could intervene during generation to avoid tokens where the model's uncertainty is primarily epistemic and the risk of hallucinations is high, or just highlight such tokens in a user interface. Its predictions could also be used to fine-tune models to avoid these tokens themselves, analogously to popular approaches for promoting desirable model behaviors with human feedback (Ziegler et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023).

For more discussion of our setup and some of its implications, we provide an FAQ in Appendix A.

Our contributions can be summarized as follows:

1. We show that small (linear) probes learn to accurately predict (AUC > 0.9) when the large model will be confident on individual tokens across model pairings and datasets. See Figure 1 for details.
2. We show that heads trained to disentangle uncertainty on text from one domain—in our case, Wikipedia articles—transfer to others, like code (AUC > 0.8). This suggests that the heads are not simply learning to rely on domain-specific token correlations or other

[1]This is not the only source of hallucinations, which can also arise from *e.g.* false information in the training set. Nevertheless, we expect that the majority can be described this way.

heuristics to achieve high accuracy but may be "reading" more robust internal representations of uncertainty present in the small model.

3. We investigate an unrelated, fully unsupervised method for the same task (Figure 2), inspired by in-context learning, and obtain non-trivial results.

2. Related work

2.1. Identifying epistemic uncertainty

Disentangling uncertainty in probability distributions output by neural networks is a longstanding and active area of research (Hüllermeier & Waegeman, 2021). Bayesian neural networks (BNNs) maintain (approximate) posteriors of the parameters of a neural network, offering a principled way to reason about model uncertainty, albeit at a prohibitive computational cost (Welling & Teh, 2011). While ensemble-based approximations of said posteriors have shown some promise (Osband & Roy, 2015; Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017), scaling these methods to modern LLMs has proven challenging (Gleave & Irving, 2022; Osband et al., 2022). On the other hand, concurrent work has shown that ensembling clarifications of the inputs for state-of-the-art LLMs—rather than multiple LLMs—provide reliable uncertainty estimates in question-answering settings (Hou et al., 2023).

Epistemic neural networks, or "epinets," are modified neural networks conditioned on an additional epistemic index that produce expressive joint predictions over each combination of classes in a classification task. Changes in the output of an epinet induced by varying the epistemic index can be used to estimate the degree of epistemic uncertainty in the output (Osband et al., 2021). Small epinets trained on the final representations of ResNets (He et al., 2016) and BERT language models (Devlin et al., 2019) can produce joint predictions for ImageNet (Deng et al., 2009) and GLUE (Wang et al., 2018) classification tasks, respectively. They have also shown promise in active learning, as they can be used to promote epistemically uncertain training examples during model fine-tuning (Osband et al., 2023). However, they have generally been evaluated in-distribution on relatively small models—at and below 100M parameters—and simple classification tasks, with limited success elsewhere (Verma et al., 2023).

Concurrent work (Johnson et al., 2024) modifies the training procedure for generative models by asking models to predict pairs and allowing it to cheat. They show an equivalence between being good at cheating and second-order calibration. They then show that second-order calibrated models can accurate estimate what they don't know, which was previously explored by Jung et al. (2020) in a theoretical setting.

Kadavath et al. (2022) directly fine-tunes large language models to predict the probability that they answer well-formed questions correctly, indirectly estimating epistemic uncertainty. The authors achieve high accuracy in-distribution and demonstrate promising trends; confidence predictions on out-of-distribution (OOD) questions are still accurate, larger models are better able to estimate their OOD uncertainty than smaller ones, and confidence predictions tend to increase as relevant "hints" are provided in-context. Lin et al. (2022) obtains similar results by fine-tuning language models to output confidence on arithmetic tasks in text. Both works study extremely large language models, up to the scale of GPT-3 (Brown et al., 2020). However, both focus on the question-answering setting, where there is one, known answer and uncertainty is effectively always epistemic. Rather than gauge how much epistemic uncertainty is present *conditioned on the fact that uncertainty at a token is primarily epistemic*, we seek to identify tokens where model uncertainty is primarily epistemic; to distinguish between tokens that are "knowable" and tokens that are not.

2.2. In-context learning

While LLMs store knowledge in their parameters (Petroni et al., 2019) and can learn new knowledge directly via fine-tuning (De Cao et al., 2021; Mitchell et al., 2022; 2021), they are also adept at learning in-context (Brown et al., 2020). Si et al. (2023); Zheng et al. (2023); Pezeshkpour (2023) demonstrate that, when relevant new information is added to prompts, LLMs can update their predictions even when said information is in conflict with model's internal knowledge. Si et al. (2023) also shows empirically that larger models are better at updating their knowledge in this way. Our unsupervised method relies on identifying when models rely most heavily on their in-context learning capabilities.

3. Setup

3.1. High-level task description

Consider a setting where we have access to M_{small}, a comparatively small but still useful language model, and M_{large}, a larger and significantly more capable but impractically expensive counterpart. For convenience, we assume in this work that the two language models share the same vocabulary $\mathcal{T}$, but this is not necessary.

At a high level, we are interested in the following task. Given a text prompt $x = \{x_i\}_{i=1}^{N}$, letting x_i be the i-th token of x and x_i^j be the substring of x between indices i and j, we wish to identify indices k where the distributions $M_{\text{small}}(x_1^k)$ and $M_{\text{large}}(x_1^k)$ are substantially dissimilar *without access to* M_{large}. Under the assumption that language models are generally well-calibrated, we are particularly interested in tokens about which the small language model

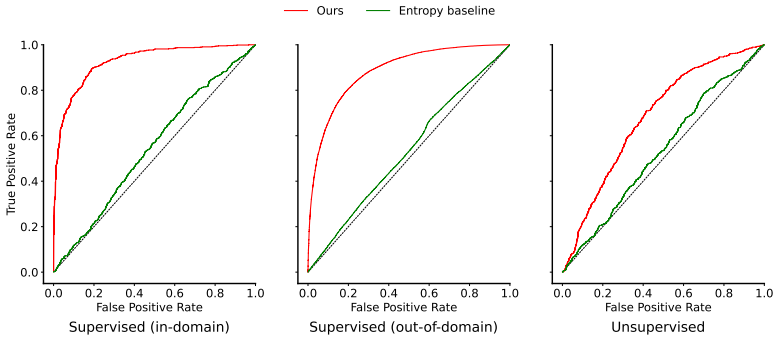

Figure 3. **High-level classification results for our supervised and unsupervised methods.** ROC curves for linear probes trained with the model pairing LLaMA 7B / LLaMA 65B, evaluated both in (*left*) and out (*middle*) of the distribution of the probes' Wikipedia training data. All classifier probes are evaluated on balanced test sets. *Right*: ROC curve for unsupervised ICLT method with the model pairing LLaMA 7B / LLaMA 65B evaluated on the Wikipedia test set.

appears uncertain (*i.e.* tokens for which predictions have large entropy). Note that a successful method for this task would in principle permit a number of useful interventions on M_{small}. For example, if during autoregressive generation using M_{small} an (accurate) prediction head indicates that M_{large} is extremely confident about a token on which M_{small} is uncertain—indicating epistemic uncertainty at that token on the part of M_{small}[2]—it may be prudent to rewind generation and resample, or simply to highlight the token in a user interface for M_{small}. While M_{large} may not itself be perfect, improving the capabilities of M_{large} can also be expected to improve the utility of such a head.

We have not yet defined "substantially dissimilar." The most granular (and practically applicable) version of this task would involve predicting the difference in probability between M_{small} and M_{large} at each token in $\mathcal{T}$, but for now we focus on a simpler variant: predicting high-level summary statistics about M_{large}'s output distribution using only intermediate representations from M_{small}. We experiment with a variety of closely related target values (see Appendix D for details), but unless otherwise noted we default to the conceptually simplest option: the Shannon entropy of the large model's prediction:

$$H(M_{\text{large}}(x)) = -\sum_{t \in \mathcal{T}} M_{\text{large}}(x)_t \log M_{\text{large}}(x)_t.$$

A low entropy value indicates that the large model is placing probability mass on a few tokens, indicating the large model is fairly certain about its next token prediction. In this case, we consider that the small model's uncertainty is "epistemic-like." On the other hand, a high value indicates that the large model is also uncertain and we consider the small model's uncertainty to be "aleatoric-like." In Figure 4, we visualize differences in paired model entropies, and in Appendix I, we provide examples of labeled tokens in real data.

There are clear limitations to our framing, most importantly that our "large" language models still exhibit epistemic uncertainty in their own right, which introduces label noise. We also ignore sequence-level, semantic uncertainty (Kuhn et al., 2023) and mixtures of epistemic and aleatoric uncertainty.[3] Nevertheless, solving this narrower but still non-trivial problem is a meaningful first step. We answer some common questions about the setup in Appendix A.

3.2. Models

While we stress again that our task setup does not necessarily require that both the small and large models share the same vocabulary, for convenience, we consider "hierarchies" of open-source language models like LLaMA (Touvron et al., 2023a), Llama 2 (chat and non-chat variants) (Touvron et al., 2023b), and Pythia (Biderman et al., 2023). Within each family, vocabulary, architecture, and training data are shared, allowing for simple evaluation across model sizes and, in the case of Pythia, training checkpoints. LLaMA models are available at 7B, 13B, 33B, and 65B parameters, Llama 2 at 7B, 13B, and 70B (with and without instruction tuning), and Pythia at 70M, 160M, 410M, 1B, 1.4B, 2.8B, 6.9B, and 12B (along with intermediate checkpoints for each).

3.3. Datasets

For all experiments, we use free-form text data not in the training data of the small and large models in each pair. It is specifically important that the large language model not have encountered and memorized the text in question; otherwise, we wouldn't expect to be able to predict the large model's

[2]Barring undue training set memorization by the large language model.

[3]Granted, it could be argued that it is often possible to pinpoint one token where a model ultimately commits to an output it is conceptually uncertain about.

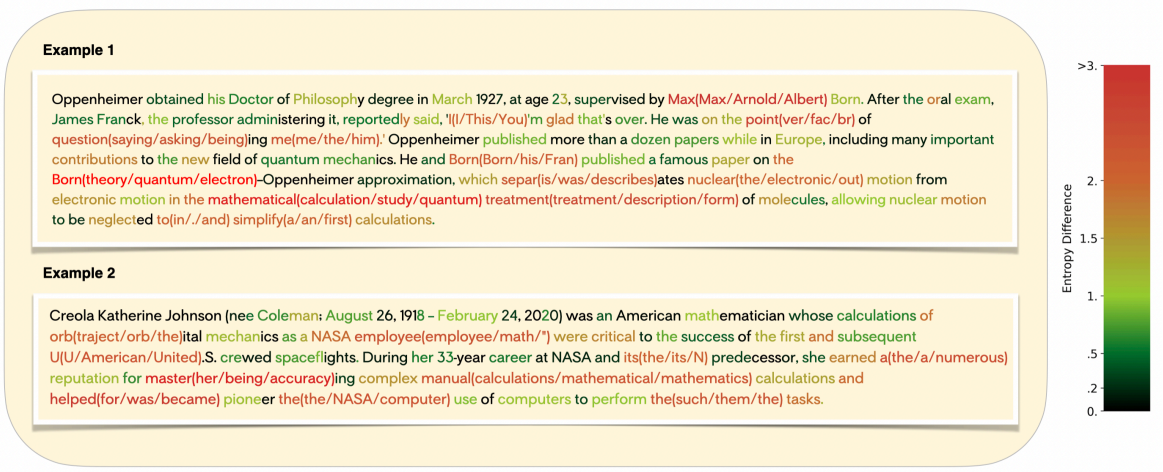

Figure 4. **Given a knowledgeable enough "large" model, tokens where a small model is unconfident while the large model is confident can be thought of as instances of epistemic uncertainty on the part of the small model.** Snippets from Wikipedia articles with tokens highlighted wherever the conditional predictive entropies of a pair of small and large language models (LLaMAs 7B and 30B) differ. For tokens where the entropy difference is larger than 2 bits, we also display the small model's top 3 token predictions in parentheses. Qualitatively, we observe that tokens at which the large model has near-zero entropy but the small one does not tend to be "epistemic" in nature, corresponding to *e.g.* dates, people, and specific technical vocabulary. See Appendix C for an autoregressive version of the same.

confidence without specific knowledge of its training set. Ideally, because we are most interested in small model uncertainty attributable to content rather than just text format, we also prefer data in a format familiar to both models. For LLaMA models, we use the set of Wikipedia articles created (*not* last edited) between the models' training cutoff and June 2023. The training set for the LLaMA models contains a small fraction of older Wikipedia data (Touvron et al., 2023a;b). We also use the designated Pile evaluation and test sets (Gao et al., 2021). Each Pile subset can be further subdivided into the Pile's component parts. In our out-of-distribution evaluations, we focus on three: "Code," containing code data from GitHub, "Stack Exchange," data from Q&A forums, and "EuroParl," a smaller subset of multilingual European Parliament proceedings.[4] For samples from each set, see Section J in the appendix.

3.4. Baselines

Correlations between the predictions of the large and small models make it possible to predict the entropy of the large model's prediction simply by outputting the entropy of the small model's prediction (see Figure 5). Similarly, certain tokens are correlated with high- or low-entropy predictions for the following token. Any model will often have high entropy after a period, for example. While we filter our data to mitigate these effects and ensure that our heads are learning nontrivial functions, as sanity checks, we also include the following baseline methods wherever relevant:

[4]Corresponding to the `pile_set_name` `Github`, `StackExchange`, and `EuroParl`, respectively.

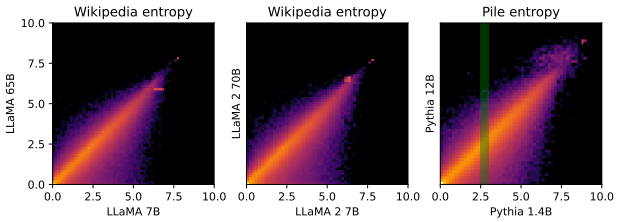

Figure 5. **Small model entropy and large model entropy are heavily correlated.** Heatmaps (with log-scale color schemes) of the entropies of token probabilities output by smaller and larger language models. Across model types and datasets, these values are heavily correlated. To rule out that our heads learn to rely on this fact, we train separate heads on tokens from narrow bands of the small model's predictive entropy (example band highlighted in green) within which the correlation is negligible.

Best entropy threshold (BET): For our binary classification tasks, we list the accuracy of the best possible classifier that outputs labels based on a threshold of the small model's entropy.

Best entropy threshold after finetuning (BET-FT): To separate out the effects of the fact that our heads are trained on data that is slightly out of the corresponding language

model's training distribution, we finetune the small language model's final language modeling layer on distributions predicted by the large model and repeat the BET benchmark.

Small model entropy (SME): Analogously, for our regression tasks, we list the error of simply predicting the entropy of the small model's predicted distribution.

Prediction from initial embeddings (PIE): As an input to our classifier and regression heads, we replace the final embedding output by the small model with the initial embedding in the small model, before any transformer layers. This can only be effective if it is possible to predict the large model's confidence from the preceding token alone.

Neural Net Ensemble (NNE) We also compare our supervised method with existing uncertainty quantification techniques. NN Ensemble (Lakshminarayanan et al., 2017) trains multiple models in parallel and uses mean and variance of the models' prediction to estimate uncertainty. Training multiple LLMs such as LlaMA from scratch may be computational infeasible, we instead finetune the small language model's final language modeling layer using LoRA (Hu et al., 2021) with 14 different random seeds. We then use the mean and variance of the entropy of the finetuned models' prediction for the binary classification task.

4. Supervised experiments

4.1. Binary classification (with gap)

Setup: We begin with the "easiest" variant of the task: identifying tokens for which the large model's entropy is close to zero.

As previously noted, an important consideration in our experiments is that the entropies of the predicted distributions from the small model and the large model tend to be heavily correlated, meaning that it is possible to trivially perform well at tasks depending on predicting the entropy of the large model's distribution simply by *e.g.* computing the entropy of the small model's prediction directly. For that reason, we try training separate classifiers for different "bands," or ranges of the values of the small model's prediction's entropy (*e.g.* one dedicated classifier for all tokens where the small model's next token prediction has entropy between 2 and 3). The narrower the bands, the weaker the aforementioned correlation within each band.

We train both unconditional classification heads and specialized heads for individual bands. Heads take as input next token embeddings from S. We find embeddings from the middle layers of each model are best (see Appendix F.6 for more details), though different layers do sometimes perform better out of distribution. To eliminate error caused by ambiguous tokens near bin boundaries and establish a clearer proof of concept, we initially focus on a binary classification task where we introduce an artificial gap between bins, reducing the task to predicting whether, conditioned on the small model's prediction having entropy inside a narrow band (*e.g.* $[k, k+1]$ for some k), the large model predictive entropy is very low or high, *i.e.* close to zero ($\lessapprox 0.2$) or within the same narrow band. For precise details on the binary classification setup, see Appendix F.1. For all classification tasks, we heavily filter tokens to 1) balance class labels and 2) equalize counts of the token immediately before the target token across classes to eliminate trivial correlations. Note that both interventions make the task more difficult.

Transfer to unseen distributions: Including this filtering process, we have taken several measures to prevent our heads from learning overly simplistic functions. While we believe these measures are sufficient to elicit non-trivial behavior in practice, they do not rule out that our heads learn to depend on heuristic, domain-specific cues, like correlations between class labels and *2*-grams in the prompt, rather than generalizable information computed internally by the small model.[5] To determine whether this occurs, we perform additional evaluations on out-of-distribution data. For example, we evaluate heads trained exclusively on tokens from Wikipedia articles on code data from the Pile.

Selected results for the binary variant are given in Table 1. Additional experiments, including results for different entropy bands and model pairings, are described in Appendix F.3. It is important to note that, because class labels are determined by the large model and we balance training and evaluation sets by class, each classifier is trained and evaluated on slightly different subsets of the tokens in its respective dataset.

In general, with minimal hyperparameter tuning, it is possible to train accurate linear classifiers (AUC > 0.9) across model pairings and datasets. These classifiers perform nearly as well outside of their training distribution, on code, multilingual, and Q&A-style evaluations.

4.2. Binary classification (without gap)

The results in the previous section clearly demonstrate that embeddings from the small model contain enough information to distinguish between tokens where the predicted distribution output by the large model has high or near-zero entropy, but the gap introduced between bins means the resulting classifiers cannot be used in practice. Here, we train binary classifiers without a gap. As before, we use entropy bands and balance classes with equalized previous token counts. We set the boundary between bins to 1 bit (somewhat surprisingly, the choice of threshold does not

[5] One could imagine an internal "flag," loosely defined, denoting the presence of epistemic uncertainty.

Table 1. **Linear classifiers of small model activations can reliably predict when the large model is confident, both in and out of distribution.** AUROC of binary classifiers for large model predictive entropy (with an artificial gap) all trained on the Wikipedia set. Inputs include tokens for which the small model's predictive entropy is in the range [2, 3). Labels correspond to whether the large model's predictive entropy is 1) near zero or 2) within the same band. Training and test sets are all class- and token-balanced to mitigate trivial entropy correlations.

MODEL	S	L	TYPE	TEST SET	AUC
LLAMA	7B	30B	MLP	WIKIPEDIA	**0.94**
LLAMA	7B	30B	LINEAR	WIKIPEDIA	**0.94**
LLAMA	7B	30B	BET	WIKIPEDIA	0.54
LLAMA	7B	30B	BET-FT	WIKIPEDIA	0.63
LLAMA	7B	30B	NNE	WIKIPEDIA	0.64
LLAMA	7B	65B	MLP	WIKIPEDIA	**0.93**
LLAMA	7B	65B	LINEAR	WIKIPEDIA	**0.93**
LLAMA	7B	65B	BET	WIKIPEDIA	0.54
LLAMA	7B	65B	BET-FT	WIKIPEDIA	0.66
LLAMA	7B	65B	NNE	WIKIPEDIA	0.60
PYTHIA	1.4B	12B	MLP	WIKIPEDIA	**0.91**
PYTHIA	1.4B	12B	LINEAR	WIKIPEDIA	**0.86**
PYTHIA	1.4B	12B	BET	WIKIPEDIA	0.59
PYTHIA	1.4B	12B	BET-FT	WIKIPEDIA	0.75
LLAMA	7B	30B	LINEAR	CODE	**0.82**
LLAMA	7B	30B	LINEAR	EUROPARL	**0.79**
LLAMA	7B	30B	LINEAR	STACK EX.	**0.88**
LLAMA	7B	65B	LINEAR	CODE	**0.79**
LLAMA	7B	65B	LINEAR	EUROPARL	**0.81**
LLAMA	7B	65B	LINEAR	STACK EX.	**0.88**
PYTHIA	1.4B	12B	LINEAR	CODE	**0.75**
PYTHIA	1.4B	12B	LINEAR	EUROPARL	**0.68**
PYTHIA	1.4B	12B	LINEAR	STACK EX.	**0.80**

meaningfully affect the performance of the classifiers). Full results are provided in the appendix in Table 4.2.

As expected, points near the boundary cause the accuracy to drop—to *e.g.* approximately 75% for the 7B/65B LLaMA experiment in the [2, 3) band, down from the high 80s—but the classifiers still outperform both baselines by large margins. We experiment with the more difficult but more flexible approach of training regressions to predict the large model's predictive entropy directly in Appendix G.

5. Unsupervised experiments

Training our previous methods depends on access to a larger language model. Of course, the larger model is only a proxy for the true distribution of the training data, and one may not always have access to such a model. We have already demonstrated in Section 4.1 that small (linear) heads can be trained on the small model's embeddings to classify whether the larger model's predictive entropy would be very low or high. In this section, we attempt to elicit such information from the small model directly, and without any additional training.

Given a prompt p, we generate variations of the prompt for in-context learning using the small model's top k token predictions t_i, with $i \in [1, k]$. Specifically, separately for each $i \in [1, k]$, we prepend "$p + t_i$" to the original prompt. We then feed this "repeated prompt" back into the model for next-token prediction and measure the degree to which the model repeats information in its provided "hint." Note that the resulting prompts are repetitive and often highly nonsensical; we do not even complete words in cases where the "hint" token is merely a word fragment. Empirically, our method is not sensitive to our choice of how the relevant context is provided (see Appendix H.2 for an ablation study). See Figure 2 for an illustration of the method, which we call the In-Context Learning Test (ICLT).

The intuition behind our unsupervised method comes from in-context learning and model's second-order calibration. LLMs have demonstrated remarkable capabilities to learn from context (Brown et al., 2020). However, existing work typically focuses on using model's in-context learning capabilities to extract latent knowledge from LLMs (Petroni et al., 2019; Burns et al., 2023) or instill new knowledge during inference time (Si et al., 2023; Zheng et al., 2023; Pezeshkpour, 2023). Here, we hypothesize that LLMs are multi-calibrated - meaning that they can not only make predictions for the *expectation* of next token, but also for *higher moments* of the next token distribution (Jung et al., 2020). Being (second-order) multi-calibrated implies that models have accurate uncertainty estimation about its predictions (Jung et al., 2020; Johnson et al., 2024). Here, we further speculate that models learn differently from their contexts in the presence of different types of uncertainty. Specifically, we hypothesize that they are more likely to lean on information in their contexts if they are epistemically uncertain and less likely otherwise.

We first design a toy experiment to demonstrate that such "selective in-context learning" capabilities are possible for transformer architectures at all. We then try out our unsupervised method on real-world LLMs.

5.1. Synthetic proof of concept

Setup: To build intuition for our method, we first consider a simplified synthetic task. We construct $\langle$question, answer$\rangle$ pairs where each question consists of 1) a single bit indicating whether it's epistemic (0) or aleatoric (1) and 2) bits uniquely identifying the question. The answer is a single bit. For "epistemic" questions, answers are drawn uniformly at random once at initialization and then fixed (and can therefore be memorized). For "aleatoric" questions, answers are

resampled uniformly at random every time the question is encountered. A sample question/answer pair is given below:

$$\langle \underbrace{0}_{\textit{epistemic}} \ \underbrace{00101011101011}_{\textit{question index}} \ \underbrace{1}_{\textit{answer}} \rangle$$

We train a small ($\sim$100M-parameter) language model to predict the answer to questions in the k-shot setting, choosing hyperparameters such that target questions are occasionally duplicated as examples in the model's prompt, permitting "copying" (details in Appendix H). We train the model until convergence and examine the model's in-context learning behavior on both types of questions. We expect the model to learn that the nature of its uncertainty for any given question is uniquely determined by the first bit. As a result, the model should perform in-context learning and update its predictions for epistemic questions when prompted with applicable information in the same context. On the other hand, the model should not copy information from its prompt if the question is aleatoric and the model's predicted distribution over the answer choices is correct.

Results: In Figure 6, upper-right panel, we showcase model behavior for two example questions, one epistemic and one aleatoric. We first prompt the model with the question without any additional context. In both cases, the model initially places approximately the same probability on both answers. We then prepend a copy of the question and an answer bit to the original prompt and re-run the model to test whether it increases the likelihood of the provided answer. We observe that the model consistently performs in-context learning to answer epistemic questions, regardless of the correctness of the provided answer, but does not change its predictions in the aleatoric case. Similar knowledge-updating behavior is pointed out by (Si et al., 2023).

5.2. ICLT on real data

In the synthetic experiment, to form an internal representation of uncertainty, the model only needs to observe that the epistemic/aleatoric nature of its uncertainty is uniquely determined by the first bit in the question. Real language is obviously less clear-cut. However, success in the synthetic setup hints that language models can form internal representations of different types of uncertainty and adjust their behavior accordingly. If real language models are capable of the analogous task of classifying natural-language prompts (even in a "soft" way), we may be able to extract that information in a similar fashion: by simply measuring how "suggestible" the language models are under each prompt. To quantify the result, we look at whether the model significantly increases the likelihood of repeating the token t_i when it is presented in the context compared to its likelihood in the original generation. In the lower two panels of Figure 6, we showcase two examples (one epistemic and one aleatoric). The observed behavior agrees with our intuition

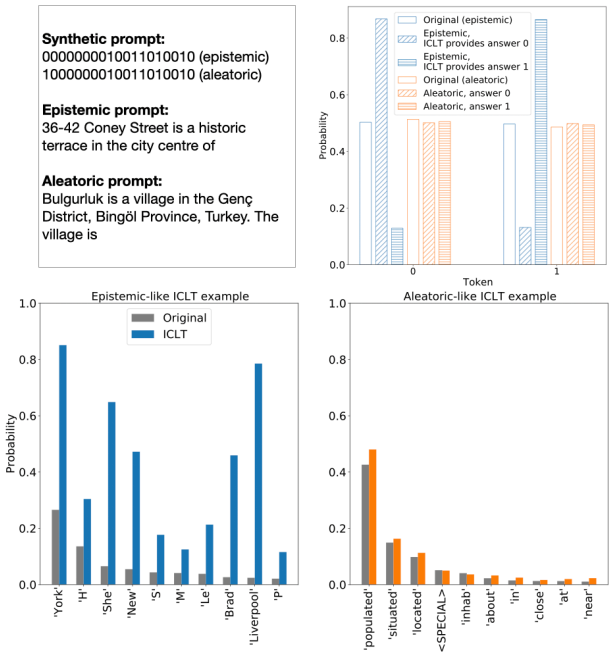

Figure 6. **Concrete examples of ICLT in synthetic and empirical settings.** *Top right:* Synthetic results. *Bottom:* Empirical results using LLaMA 7B. Gray bars represent the model's original prediction without additional in-context information. Blue and orange bars are the model's predicted probability for token i conditioned on a repeated prompt containing token i.

for ICLT. Note that these are cherry-picked; in Appendix H.1 we list more diverse examples and speculate why ICLT works less well in some cases.

For a larger-scale evaluation, we employ a larger model to identify epistemic uncertainty, as in Section 4.1. We perform ICLT with top k tokens as context and use the minimum entropy produced by the model when prompted with the top k tokens as additional context. We formulate the binary classification task as before, with token-based filtering and a gap between the two bins. Figure 7 shows the original entropy for LLaMA 7B, the entropy prediction for LLaMA 65B, and the minimum entropy across repetitions from the ICLT method. Minimum entropy is a natural choice based on our intuition behind the ICLT method, and in Appendix H.2 we include further discussion on the choice of metric. We vary both large and small model sizes and report results in Table 2, where we compare the ICLT performance with the BET baseline. See Table 12 in Appendix H for Llama2 results.

The intuition behind inserting a separator token after the context, before we re-prompt the model comes from the idea that the next token contains epistemic uncertainty, the relevant information should be present in multiple documents

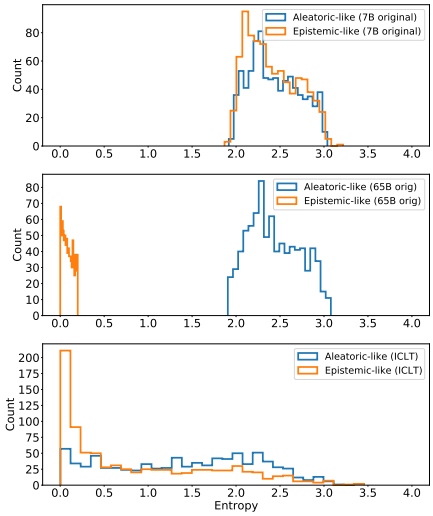

Figure 7. **ICLT separates epistemic from aleatoric tokens.** *Top:* Original predictive entropy of LLaMA 7B for tokens in the class-balanced Wikipedia test set (used for BET baseline). *Middle:* Predictive entropy of the 65B model (source of aleatoric/epistemic labels). *Bottom:* Minimum predictive entropy of LLaMA 7B during ICLT. Relative to aleatoric ones, entropy tends to decrease on epistemic examples when a hint is provided in context.

in the training data. As a result, if we were to train the small model longer, such uncertainty could be reduced by learning the relevant information. By inserting a document separator, we are simulating the case when an additional document is present the context, providing the relevant information. In contrast, if the token contains aleatoric-like uncertainty, we should expect that the true language distribution also contains large uncertainty and information provided from a distinct document, the small model will not in-context learn any information. In Table 3 we examine the impact of the use of document separator and observe that the use of a separator token is crucial in the performance of the ICLT method. Given the importance of separator token in the ICLT method, we include a preliminary discussion in Appendix H.3 on why it might related to the failure case on Pythia.

Table 2. **ICLT results: LLaMA.** "Band" denotes lower and upper bounds of small model predictive entropy using LLaMA models of various sizes.

S	L	Band	Baseline		Repetition	
			AUC	Acc	AUC	Acc
7B	30B	$[2.0, 3.0)$	0.56	0.55	0.71	0.67
7B	30B	$[3.0, 4.0)$	0.52	0.55	0.71	0.66
7B	65B	$[2.0, 3.0)$	0.56	0.55	0.68	0.66
7B	65B	$[3.0, 4.0)$	0.54	0.54	0.70	0.68
30B	65B	$[2.0, 3.0)$	0.55	0.54	0.61	0.60

Table 3. **ICLT results: Ablation on token separator** Ablation study on different types of separators used in between the context and the prompt. The separator is important in the performance of ICLT method. Small model: LLaMA 7B, Large model: LLaMA 30B, entropy band: [2.0, 3.0)]

Separator Used Type	AUC	Acc
Original ICLT (BOS)	0.68	63.4
BOS and EOS	0.67	64.0
EOS only	0.62	57.5
None	0.56	52.4

Table 4. **ICLT results: Pythia (failure case).** "Band" denotes lower and upper bounds of small model predictive entropy using Pythia models of various sizes. We use the entropy band $[2.0, 3.0)$.

S	L	Count	Baseline		Repetition	
			AUC	Acc	AUC	Acc
70M	12B	1025	0.60	0.52	0.54	0.52
410M	12B	1927	0.59	0.52	0.53	0.52
1B	12B	2314	0.54	0.54	0.54	0.54

6. Conclusion

In this paper, we demonstrate that 1) across a variety of text domains and model sizes, LLM embeddings contain enough information to ascertain the certainty of more capable models and 2) this information is sometimes correlated with how willing the model is to copy information in its prompt, permitting unsupervised prediction. In this sense, at least, LLMs "know what is *knowable*," distinguishing between prompts for which there is effectively just one correct answer and prompts for which there are many.

Our work is preliminary, and problems remain to be solved before these techniques can be incorporated into practical systems to *e.g.* reduce hallucinations. Some are straightforward engineering challenges; future work is needed to assess performance on more model pairings and datasets,[6] tune the number of "entropy bands" and thresholds, understand qualitatively where our classifiers fail, and improve classifier performance (both precision and recall) on heavily unbalanced real-world token datasets. Others require more conceptual work. For our initial assumptions to hold, the large models in our pairings should exhibit as little epistemic uncertainty as possible. Whether this goal simply requires increasing the scale of the large model and choosing the right dataset or can be accomplished by other means—perhaps by applying our unsupervised approach to the larger model before using it to generate labels for supervised probes—is currently unclear.

[6] In particular, we are interested in measuring how performance evolves as the larger model is increased in size.

Acknowledgements

We thank Sham Kakade and Garrett Tanzer for useful discussions and feedback on the manuscript.

GA is supported by a fellowship from the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. NV acknowledges funding from NSF grant DMS-2134157 and DOE grant DE-SC0022199. BB is supported by a Simons Investigator Fellowship, NSF grant DMS-2134157, DARPA grant W911NF2010021, and DOE grant DE-SC0022199. BB is currently also affiliated with OpenAI, but this work was done while he was at Harvard. BE acknowledges funding from the NSF Graduate Research Fellowship Program under award DGE-214074, the ONR under award N00014-22-1-2377, and the NSF under award IIS 2229881. This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

Impact Statement

Our study on identifying different types of uncertainty in language models carries broader impacts on enhancing AI trustworthiness and AI safety. A key future application of distinguishing between aleatoric and epistemic uncertainty is in reducing hallucinations, which improves the truthfulness of LLM outputs, and increases the reliability of AI in critical applications.

References

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Cole, J. R., Zhang, M. J. Q., Gillick, D., Eisenschlos, J. M., Dhingra, B., and Eisenstein, J. Selectively answering ambiguous questions, 2023.

De Cao, N., Aziz, W., and Titov, I. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Desai, S. and Durrett, G. Calibration of pre-trained transformers. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 295–302, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.21. URL https://aclanthology.org/2020.emnlp-main.21.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:52967399.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027, 2021. URL https://arxiv.org/abs/2101.00027.

Gleave, A. and Irving, G. Uncertainty estimation for language reward models, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022.

Hou, B., Liu, Y., Qian, K., Andreas, J., Chang, S., and Zhang, Y. Decomposing uncertainty for large language models through input clarification ensembling. *arXiv preprint arXiv:2311.08718*, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021.

Huang, B. R. and Kwon, J. Does it know?: Probing for uncertainty in language model latent beliefs. In *NeurIPS Workshop on Attributing Model Behavior at Scale*, 2023. URL https://openreview.net/forum?id=uSvN2oozRK.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, pp. 457–506, 2021. doi: 10.1007/s10994-021-05946-3.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023. ISSN 1557-7341. doi: 10.1145/3571730. URL http://dx.doi.org/10.1145/3571730.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. 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, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57.

Johnson, D. D., Tarlow, D., Duvenaud, D., and Maddison, C. J. Experts don't cheat: Learning what you don't know by predicting pairs, 2024.

Jung, C., Lee, C., Pai, M. M., Roth, A., and Vohra, R. Moment multicalibration for uncertainty estimation, 2020.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., and Kaplan, J. Language models (mostly) know what they know, 2022.

Kamath, A., Jia, R., and Liang, P. Selective question answering under domain shift. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5684–5696, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.503. URL https://aclanthology.org/2020.acl-main.503.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=aLLuYpn83y.

Lin, S., Hilton, J., and Evans, O. Teaching models to express their uncertainty in words, 2022.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. On faithfulness and factuality in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL https://aclanthology.org/2020.acl-main.173.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Mitchell, E., Lin, C., Bosselut, A., Manning, C. D., and Finn, C. Memory-based model editing at scale. In *International Conference on Machine Learning*, pp. 15817–15831. PMLR, 2022.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. GPT-4 technical report, 2023.

Osband, I. and Roy, B. V. Bootstrapped thompson sampling and deep exploration, 2015.

Osband, I., Wen, Z., Asghari, M., Ibrahimi, M., Lu, X., and Roy, B. V. Epistemic neural networks. *CoRR*, abs/2107.08924, 2021. URL https://arxiv.org/abs/2107.08924.

Osband, I., Wen, Z., Asghari, S. M., Dwaracherla, V., Hao, B., Ibrahimi, M., Lawson, D., Lu, X., O'Donoghue, B., and Roy, B. V. The neural testbed: Evaluating joint predictions, 2022.

Osband, I., Asghari, S. M., Roy, B. V., McAleese, N., Aslanides, J., and Irving, G. Fine-tuning language models via epistemic neural networks, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, pp. 8026–8037. Curran Associates Inc., Red Hook, NY, USA, 2019. doi: 10.5555/3454287.3455008.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

Pezeshkpour, P. Measuring and modifying factual knowledge in large language models. *arXiv preprint arXiv:2306.06264*, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization:

Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Si, C., Gan, Z., Yang, Z., Wang, S., Wang, J., Boyd-Graber, J., and Wang, L. Prompting gpt-3 to be reliable, 2023.

Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Fine-tuning language models for factuality, 2023a.

Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. D. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback, 2023b.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Varshney, N., Mishra, S., and Baral, C. Investigating selective prediction approaches across several tasks in IID, OOD, and adversarial settings. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1995–2002, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.158. URL https://aclanthology.org/2022.findings-acl.158.

Verma, S., Tran, K., Ali, Y., and Min, G. Reducing llm hallucinations using epistemic neural networks, 2023.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupała, G., and Alishahi, A. (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, 2011. URL https://api.semanticscholar.org/CorpusID:2178983.

Yang, Y., Zha, K., Chen, Y., Wang, H., and Katabi, D. Delving into deep imbalanced regression. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11842–11851. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yang21m.html.

Zheng, C., Li, L., Dong, Q., Fan, Y., Wu, Z., Xu, J., and Chang, B. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences, 2020.

Appendix

Table of Contents

A FAQ **20**

B Related work (extended) **23**
B.1 Measuring calibration . . . 23
B.2 Combating hallucinations . . . 23

C Figure 4 with auto-regressive generation **24**

D Training objectives **25**

E Dataset details **26**

F Classification (extended) **27**
F.1 Binary classification setup (with gap) . . . 27
F.2 Binary classification setup (without gap) . . . 27
F.3 Binary classification full results (gap) . . . 27
F.4 Binary classification full results (without gap) . . . 27
F.5 Inter-run variance . . . 29
F.6 Transfer between models . . . 29
F.7 Choice of embedding . . . 29
F.8 Choice of checkpoint . . . 30

G Regressions **35**

H Unsupervised classification (extended) **37**
H.1 Additional ICLT results . . . 37
H.2 Ablation study on context provided for ICLT . . . 39
H.3 Why does ICLT fail on Pythia? . . . 40

I Labeled examples **41**

J Dataset samples **44**

A. FAQ

In this section, we answer some common questions about our approach.

What can be expected to happen as you increase the size of the "large" model?

A natural question to ask is how the difficulty of the uncertainty classification task is affected by the size of the "large" model. In the paper, we provide results using Pythia 12B as the "large" model as well as results for the LLaMA pairings 7B/30B and 7B/65B. For both the supervised and unsupervised experiments, there does not appear to be a substantial difference between the two sets of LLaMA figures, either in or out of domain. Supervised results are weaker for the Pythia heads, albeit that those also rely on a smaller "small" model (Pythia 1.4B) than the LLaMA heads. While this might simply imply that the 30B LLaMA model is already sufficiently "knowledgeable" for our purposes, there are also countervailing reasons to believe that the prediction task grows both easier and more difficult along with the size of the large model.

On the one hand, a larger and better-trained model knows more, and can be expected to have less epistemic uncertainty over the same dataset than a smaller and less capable model. As a result, one can expect to find less noise in the label sets generated by the large model; specifically, there will be fewer tokens in the "high entropy" category that actually correspond to "knowable" facts. We expect this to simplify the task.

On the other hand, what constitutes a low-entropy "fact" for a model may become more subtle as it grows; larger models are better at noticing patterns in text that might at first glance appear "aleatoric" (*e.g.* obscure idioms, archaic verb conjugations, and other things we as human readers might not even be aware of). Depending on the makeup of their training sets, larger models can also be expected to have memorized more text, which is potentially another source of noise in the other direction. A memorized "aleatoric" passage cannot be distinguished from an instance of epistemic uncertainty using our method, and it is naturally impossible for the small model to predict whether any given passage was memorized by the larger model.

It is worth noting too that the predictions of smaller "large" models are closer to those of their respective "small" models, improving the quality of trivial baselines based on the small model's prediction. In the limit, where the small model and the large model are the same, it is trivially possible to predict the entropy of the "large" model with perfect accuracy.

We are interested in testing our methods with even larger and more knowledgeable "large" models (at the scale of *e.g.* ChatGPT) to pin down the net effects.

Why would models trained for next token prediction have internal representations of uncertainty?

Recent work in mechanistic interpretability has identified a number of useful internal representations in large language models, like a "truthfulness" direction in activation space (Burns et al., 2023; Li et al., 2023). In some cases, the intuitive benefits of these mechanisms for next token prediction is clear; internet text, for example, is more often than not correlated with true facts about the world, and so it is in principle helpful for the model to understand the difference between facts and falsehoods. Why it could be useful to form internal representations of different forms of uncertainty is less clear.

One simple explanation involves the fact our unsupervised method depends on: tokens that are more or less "epistemic" might correspondingly rely more or less on information earlier on in the prompt. Of course, the entropy of tokens in natural language is not bimodal like that of answer tokens in our synthetic setting. Nevertheless, strictly "factual" tokens do exist, and some of the same logic could apply in a "soft" way for more ambiguous real tokens.

Is it possible to generate labels without access to a "large" model?

In an attempt to permit evaluation with gold-standard labels for aleatoric and epistemic uncertainty, rather than labels based on discrepancies between small and large language models, we experimented with dataset of synthetic prompts using Wikidata, a knowledge graph derived from Wikipedia. For several hand-picked "many-to-one" data categories, containing e.g. book/author or entity/location pairs, we manually wrote a small number of few-shot template prompts *in both directions*. In the "forward" case, the model is asked to provide the answer to a question for which it is expected that there is exactly one correct answer (*e.g.* identifying the author of a book). In the "backward" case, the prompts are flipped, and the right answer is one of many (*e.g.* identifying a book written by an author). We filtered the pairs such that each "backward" prompt has at least five answers in Wikidata. "Forward" prompts are intended to elicit epistemic uncertainty, and "backward" aleatoric. See Figure 8 for examples of both.

We did not succeed at producing a diverse enough set of prompts and had issues getting the LLMs to behave as expected on the ones we did write before publication, but we still believe such a dataset could be a useful contribution. We expect that

more careful filtering (*e.g.* removing prompts for which a particular "large" LLM yields the wrong answer), more specific subcategorization (Wikidata categories tend to be extremely broad, to the point where it is difficult to write natural prompts in both directions), better prompt construction (we tried some simple few-shot settings, but not exhaustively), and potentially leveraging LLMs to produce a larger and more diverse set of prompts would yield improved results.

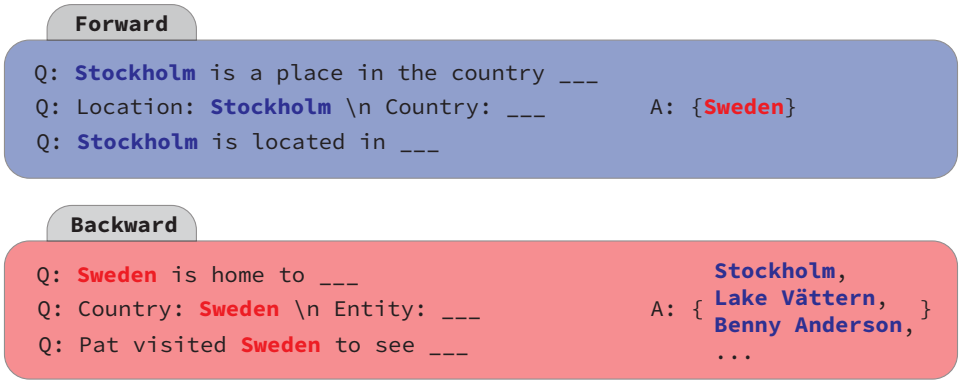

Figure 8. Examples of prompts for the "country" Wikidata category. "Forward" prompts have one correct answer; "backward" prompts have many.

Why do baseline results differ from model pairing to model pairing?

The dataset construction process depends on the entropies of both the small model (via entropy banding) and the large model (via label balancing). As such, the makeup of each test set differs slightly from experiment to experiment.

Do the two classes in the binary classification experiments really correspond to aleatoric and epistemic uncertainty?

The second (higher) bin in our classification experiments is really a catch-all bin for tokens with strictly aleatoric uncertainty, tokens with epistemic uncertainty *and* aleatoric uncertainty (on the part of both models), and tokens where both the large and the small model have significant amounts of epistemic uncertainty. This is one of the reasons why it is important to use a high-quality large model.

B. Related work (extended)

B.1. Measuring calibration

Substantial recent work has sought to answer whether large language models are well calibrated in various text domains (Desai & Durrett, 2020; Jiang et al., 2021; Kadavath et al., 2022). Calibrating language models is closely related to but not identical to the task of distinguishing between different forms of uncertainty, as we have discussed. For example, a model can achieve perfect calibration on balanced multiple choice questions without providing any useful information about its confidence by outputting a uniform distribution over answer choices. Nevertheless, in practice, well-calibrated models can indirectly signal confidence, *e.g.* by predicting confidence scores in text form (Kadavath et al., 2022; Tian et al., 2023b), and they can be useful for many of the same downstream applications, like selective answering (Kamath et al., 2020; Varshney et al., 2022; Cole et al., 2023).

B.2. Combating hallucinations

While robust predictors of epistemic uncertainty can in principle be used to combat factual hallucinations in language model outputs, methods that do not explicitly rely on uncertainty distinction have also proven effective. Language models can be prompted to be more truthful (Si et al., 2023), fine-tuned to output sentences like sentences identified by repeated samples of the model itself to be truthful (Tian et al., 2023a), and modified at inference time to output distributions more consistent with internal truth representations (Burns et al., 2023; Li et al., 2023). Given that our method tends to detect "correctable" tokens—tokens where a small language model is significantly less confident than a larger language model—in imbalanced data with high precision but low recall, many of these approaches could be complementary to ours, regardless of the underlying mechanism by which our method achieves high accuracy.

(Burns et al., 2023) proposes Contrast-Consistent Search (CCS), an unsupervised method for identifying directions in the activation space of an LLM that best separates contradictory answers to questions. These can then be used to elicit latent knowledge from the model, serving as an internal "truth" representation analogous to the "uncertainty" representation we seek in this work. In follow-up work, (Huang & Kwon, 2023) identifies and filters out low-confidence CCS predictions.

C. Figure 4 with auto-regressive generation

As a variant to Figure 4, we asked the small model (LLaMA 7B) to autoregressively generate a paragraph for a given prompt. We used a large model (LLaMA 30B) as reference model and used small model's auto-regressive generated texts as prompt. We highlighted texts when large model was confident but small model was unconfident (i.e., two models disagreed on the entropy of the next token generation.)

Figure 9. A snippet from a Wikipedia article with tokens highlighted wherever the conditional predictive entropies of a pair of small and large language models (LLaMAs 7B and 30B) differ. We show the small model's auto-regressive generation using top 1 token. For any token prediction where the models' entropy difference is larger than 2, we also show small model's next top 2 predictions in parenthesis.

D. Additional experiment details

D.1. Training objectives

We experimented with the following target values for our classification heads. While each may have unique advantages, we do not find it necessary to stray from large model entropy to achieve high accuracy on our classification tasks.

Large model entropy: The Shannon entropy of $M_{\text{large}}(x)$:

$$H(M_{\text{large}}(x)) = -\sum_{t \in \mathcal{T}} M_{\text{large}}(x)_t \log M_{\text{large}}(x)_t$$

Log(large model entropy): The natural log of the Shannon entropy of $M_{\text{large}}(x)$:

$$\log H(M_{\text{large}}(x)) = \log \left(-\sum_{t \in \mathcal{T}} M_{\text{large}}(x)_t \log M_{\text{large}}(x)_t \right)$$

Jensen-Shannon divergence: A symmetrized version of the Kullback-Liebler (KL) divergence used to quantify the similarity of two probability distributions over the same support. Letting D represent the KL divergence and $M(x) = 1/2(M_{\text{small}}(x) + M_{\text{large}}(x))$, the Jensen-Shannon divergence is defined as:

$$JSD(M_{\text{small}}(x) \parallel M_{\text{large}}(x)) = 1/2(D(M_{\text{small}}(x) \parallel M(x)) + D(M_{\text{small}}(x) \parallel M(x)))$$

Log(Jensen-Shannon divergence): The natural log of the same.

D.2. Training details

To train all supervised methods we use Adam (Kingma & Ba, 2015) and a learning rate of 10^{-5}. Heads have a hidden dimension of 2048 and either one or zero hidden layers (in the nonlinear and linear cases, respectively). Classification heads are trained with standard cross-entropy loss; regression heads with least squares. All heads are trained with early stopping based on validation loss. We use PyTorch (Paszke et al., 2019) and A100 GPUs.

E. Dataset details

In our experiments, we use a homemade Wikipedia dataset and parts of the Pile evaluation and test sets. We break them down as follows.

Of the 71586 articles in the Wikipedia set (approx. 18.5 million tokens), we set aside 2900 each for validation and testing and use the remaining articles as a training set for our prediction heads. All reported figures use the test set.

From the Pile evaluation set, (approx. 200 million tokens) we sample a validation set of 50k examples, leaving a head training set of 165,000 examples. For parity (and to reduce load on our cluster's filesystem) we subsample the Pile test set by randomly selecting 50k examples. Note that this reduced figure still represents more than 100 million tokens.

F. Classification (extended)

F.1. Binary classification setup (with gap)

By default, we filter tokens for our binary classification task as follows:

1. Given some text dataset, run the small and large models on every example and compute the entropy of each model's prediction for each token.
2. Remove tokens for which the small model's predictive entropy is outside the range $[k, k+1)$. By default, we set $k = 2$. Note that we bound this value from above because tokens with high small model predictive entropy are more likely to have nonzero large model predictive entropy (see Figure 5). We wish to prevent our heads from depending on this fact.
3. Remove tokens for which the large model entropy is either 1. outside $[0, 0.2)$ or 2. greater than some small distance δ from the small model's predictive entropy at that token. We use $\delta = 0.1$. Assign labels to each remaining token corresponding to these two cases.
4. Independently for each token $t \in \mathcal{V}$, where $\mathcal{V}$ is the models' shared vocabulary, let T be the set of unfiltered tokens for which the previous token in the ground truth is t. Balance the two classes within T by discarding random tokens from the larger class (specifically, discard tokens with label l with probability $1 - \min\{|T_0|, |T_1|\}/|T_l|$). This procedure is intended to obstruct heads from learning to depend on trivial class-token correlations.

F.2. Binary classification setup (without gap)

The setup for the gapless classification task is the same as that described in Section F.1, minus step 3.

F.3. Binary classification full results (gap)

In this section, we include the full set of experiments for the gapped binary classifiers described in Section 4.1.

To save space, we use the following shorthands for the names of datasets:

$$\begin{aligned}
\text{WIKIPEDIA} &\rightarrow \text{W} \\
\text{PILE} &\rightarrow \text{P} \\
\text{PILE (CODE)} &\rightarrow \text{P (C)} \\
\text{PILE (EUROPARL)} &\rightarrow \text{P (E)} \\
\text{PILE (STACK EXCHANGE)} &\rightarrow \text{P (SE)}
\end{aligned}$$

AUROC curves for the experiments described in Table 1 are given in Figure 10. Note that the EuroParl set is much smaller than both other two and exhibits much greater variance from classifier to classifier (as seen in *e.g.* the 7B/30B and 7B/65B panels). Also note that the choice of activations used for the Pythia OOD classifiers is suboptimal—see Table 6 for a comparison to other embeddings, which achieve much better performance on e.g. the Pile code set.

See Tables 5, 6, and 7 for LLaMA, Pythia, and Llama 2 results, respectively.

F.4. Inter-run variance

To gauge the sensitivity of the performance of our binary classifiers to training randomness, we train independent LLaMA 7B/65B gapped binary classifiers on the Wikipedia set with different seeds. We include 10 runs each for non-linear and linear classifiers. Seeds determine model initialization and the sequence of training data. For this experiment, dataset filters are held constant. Results are given in Figure 11. We observe very little variance between runs; the min and max AUC scores for the MLPs differ by just 0.002.

F.5. Transfer between models

How do classifiers trained on one model pairing fare in evaluations with different model pairings? Well.

See Table 8 for results. We evaluate 7B/30B classifiers using labels generated by the 65B model, and scores are just a few points lower across the board.

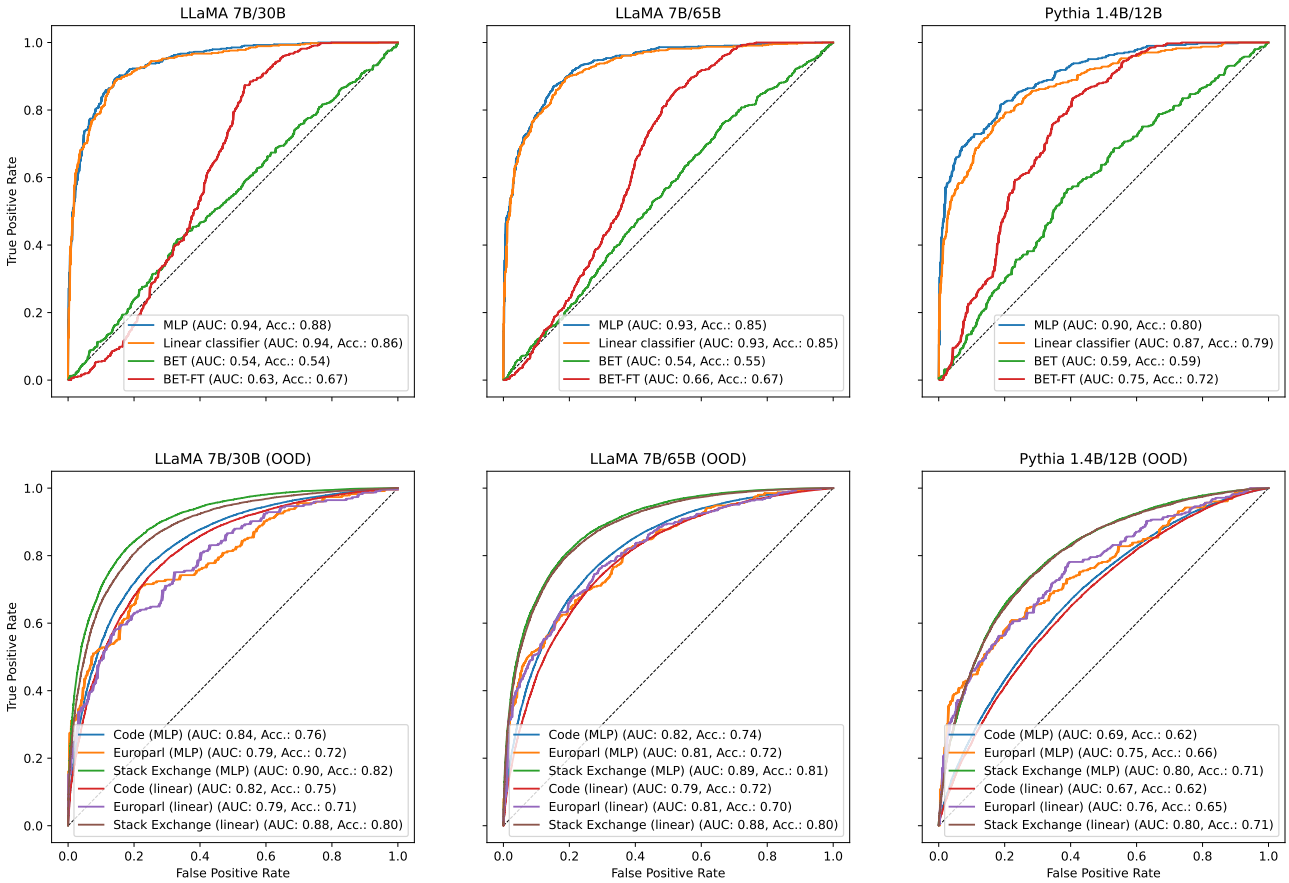

Figure 10. Visualization of the data in Table 1. *Top:* ROC curves of binary classifiers for large model predictive entropy (with an artificial gap) trained and evaluated with the class-balanced Wikipedia set. Plot titles denote small/large model pairs. Inputs include all tokens for which the small model's predictive entropy is in the range [2, 3). Labels correspond to whether the large model's predictive entropy is near zero or within the same band. Data is provided for three pairs of models (small/large). *Bottom:* The same classifiers evaluated *without additional training* on out-of-distribution data from the Pile.

F.6. Choice of embedding

We train classification heads for the Pythia 1.4B model with embeddings from different layers in each model as inputs. Pythia 1.4B has in total 17 multi-head self-attention layers. In this ablation study, we discovered heads trained on layer 1 (defining layer 17 as the "final layer" immediately before the logit) already contain enough information on different types of uncertainties in the model. Note that the results from layer 1 can be thought of as a loose "upper bound" on the performance of classifiers that learn shallow token-based heuristics as opposed to relying on internal representations in the model; while such representations are not necessarily expected to have formed by the end of the first layer, it is reasonable to expect that the probe can "see" all of the tokens in the input by that point. Binary classification results are given in Figure 12 both with linear and non-linear heads. A curious result is that representations from layer 8 seem to outperform the final embeddings we use in the main paper in both cases. Notably, performance out of distribution using embeddings from middle layers tends to be better by an even wider margin (see e.g. Table 5). We expect that more exhaustive tuning here would be fruitful.

F.7. Choice of checkpoint

We take advantage of the fact that the Pythia models were released along with intermediate checkpoints from training to test the effect of "rewinding" the small model on classification performance. Results are given in Table 9. Rewinding back to step 9000 does not have a discernible effect on the quality of the classifiers, perhaps because reducing the quality of the

Table 5. LLaMA binary classification results (with a gap) on various test sets. For training details see Section D.2, and for baseline definitions, see Section 3.4. "Layer" denotes the layer of the small model from which classifier inputs are drawn.

Model	S	L	Band	Type	Dataset (Train → Eval)	Layer	AUC	Acc
LLaMA	7B	30B	$[2,3)$	MLP	W → W	-1	0.92	0.84
LLaMA	7B	30B	$[2,3)$	Linear	W → W	-1	0.90	0.83
LLaMA	7B	30B	$[2,3)$	MLP	W → W	16	0.94	0.87
LLaMA	7B	30B	$[2,3)$	Linear	W → W	16	0.94	0.86
LLaMA	7B	30B	$[2,3)$	BET	W → W	N/A	0.54	0.54
LLaMA	7B	30B	$[2,3)$	BET-FT	W → W	N/A	0.63	0.67
LLaMA	7B	30B	$[2,3)$	NNE	W → W	N/A	0.71	0.64
LLaMA	7B	30B	$[2,3)$	MLP	P → P	-1	0.94	0.86
LLaMA	7B	30B	$[2,3)$	Linear	P → P	-1	0.92	0.85
LLaMA	7B	30B	$[2,3)$	BET	P → P	N/A	0.54	0.53
LLaMA	7B	30B	$[2,3)$	MLP	W → P (C)	-1	0.79	0.71
LLaMA	7B	30B	$[2,3)$	Linear	W → P (C)	-1	0.76	0.69
LLaMA	7B	30B	$[2,3)$	MLP	W → P (C)	16	0.84	0.76
LLaMA	7B	30B	$[2,3)$	Linear	W → P (C)	16	0.82	0.75
LLaMA	7B	30B	$[2,3)$	BET	W → P (C)	N/A	0.52	0.53
LLaMA	7B	30B	$[2,3)$	MLP	W → P (E)	-1	0.73	0.68
LLaMA	7B	30B	$[2,3)$	Linear	W → P (E)	-1	0.65	0.59
LLaMA	7B	30B	$[2,3)$	MLP	W → P (E)	16	0.79	0.70
LLaMA	7B	30B	$[2,3)$	Linear	W → P (E)	16	0.79	0.71
LLaMA	7B	30B	$[2,3)$	BET	W → P (E)	N/A	0.61	0.59
LLaMA	7B	30B	$[2,3)$	MLP	W → P (SE)	-1	0.88	0.79
LLaMA	7B	30B	$[2,3)$	Linear	W → P (SE)	-1	0.83	0.75
LLaMA	7B	30B	$[2,3)$	MLP	W → P (SE)	16	0.90	0.82
LLaMA	7B	30B	$[2,3)$	Linear	W → P (SE)	16	0.88	0.80
LLaMA	7B	30B	$[2,3)$	BET	W → P (SE)	N/A	0.54	0.54
LLaMA	7B	65B	$[2,3)$	MLP	W → W	-1	0.92	0.85
LLaMA	7B	65B	$[2,3)$	Linear	W → W	-1	0.88	0.80
LLaMA	7B	65B	$[2,3)$	MLP	W → W	16	0.93	0.86
LLaMA	7B	65B	$[2,3)$	Linear	W → W	16	0.93	0.85
LLaMA	7B	65B	$[2,3)$	BET	W → W	N/A	0.54	0.55
LLaMA	7B	65B	$[2,3)$	BET-FT	W → W	N/A	0.66	0.67
LLaMA	7B	65B	$[2,3)$	NNE	W → W	N/A	0.68	0.60
LLaMA	7B	65B	$[2,3)$	PIE	W → W	N/A	0.54	0.55
LLaMA	7B	65B	$[1,2)$	MLP	P → P	-1	0.93	0.85
LLaMA	7B	65B	$[1,2)$	Linear	P → P	-1	0.90	0.82
LLaMA	7B	65B	$[1,2)$	BET	P → P	N/A	0.55	0.54
LLaMA	7B	65B	$[2,3)$	MLP	P → P	-1	0.95	0.87
LLaMA	7B	65B	$[2,3)$	Linear	P → P	-1	0.93	0.85
LLaMA	7B	65B	$[2,3)$	BET	P → P	N/A	0.52	0.52
LLaMA	7B	65B	$[3,4)$	MLP	P → P	-1	0.95	0.87
LLaMA	7B	65B	$[3,4)$	Linear	P → P	-1	0.93	0.86
LLaMA	7B	65B	$[3,4)$	BET	P → P	N/A	0.52	0.52
LLaMA	7B	65B	$[2,3)$	MLP	W → P (C)	-1	0.78	0.69
LLaMA	7B	65B	$[2,3)$	Linear	W → P (C)	-1	0.73	0.66
LLaMA	7B	65B	$[2,3)$	MLP	W → P (C)	16	0.82	0.74
LLaMA	7B	65B	$[2,3)$	Linear	W → P (C)	16	0.79	0.72
LLaMA	7B	65B	$[2,3)$	BET	W → P (C)	N/A	0.51	0.51
LLaMA	7B	65B	$[2,3)$	MLP	W → P (E)	-1	0.73	0.64
LLaMA	7B	65B	$[2,3)$	Linear	W → P (E)	-1	0.69	0.62
LLaMA	7B	65B	$[2,3)$	MLP	W → P (E)	16	0.81	0.71
LLaMA	7B	65B	$[2,3)$	Linear	W → P (E)	16	0.81	0.70
LLaMA	7B	65B	$[2,3)$	BET	W → P (E)	N/A	0.55	0.55
LLaMA	7B	65B	$[2,3)$	MLP	W → P (SE)	-1	0.86	0.78
LLaMA	7B	65B	$[2,3)$	Linear	W → P (SE)	-1	0.83	0.75
LLaMA	7B	65B	$[2,3)$	MLP	W → P (SE)	16	0.89	0.81
LLaMA	7B	65B	$[2,3)$	Linear	W → P (SE)	16	0.88	0.80
LLaMA	7B	65B	$[2,3)$	BET	W → P (SE)	N/A	0.53	0.53

Table 6. Pythia binary classification results (with a gap) on various test sets. For training details see Section D.2, and for baseline definitions, see Section 3.4. "Layer" denotes the layer of the small model (out of 16) from which classifier inputs are drawn.

Model	S	L	Band	Type	Dataset (Train → Eval)	Layer	AUC	Acc
Pythia	1.4B	12B	$[2,3)$	MLP	W → W	-1	0.91	0.82
Pythia	1.4B	12B	$[2,3)$	Linear	W → W	-1	0.86	0.79
Pythia	1.4B	12B	$[2,3)$	MLP	W → W	8	0.90	0.81
Pythia	1.4B	12B	$[2,3)$	Linear	W → W	8	0.87	0.79
Pythia	1.4B	12B	$[2,3)$	BET	W → W	N/A	0.59	0.59
Pythia	1.4B	12B	$[2,3)$	BET-FT	W → W	N/A	0.75	0.71
Pythia	1.4B	12B	$[2,3)$	MLP	P → P	-1	0.93	0.86
Pythia	1.4B	12B	$[2,3)$	Linear	P → P	-1	0.92	0.84
Pythia	1.4B	12B	$[2,3)$	BET	P → P	N/A	0.56	0.54
Pythia	1.4B	12B	$[2,3)$	MLP	W → P (C)	-1	0.77	0.70
Pythia	1.4B	12B	$[2,3)$	Linear	W → P (C)	-1	0.75	0.67
Pythia	1.4B	12B	$[2,3)$	MLP	W → P (C)	8	0.69	0.62
Pythia	1.4B	12B	$[2,3)$	Linear	W → P (C)	8	0.67	0.62
Pythia	1.4B	12B	$[2,3)$	BET	W → P (C)	N/A	0.55	0.54
Pythia	1.4B	12B	$[2,3)$	MLP	W → P (E)	-1	0.74	0.68
Pythia	1.4B	12B	$[2,3)$	Linear	W → P (E)	-1	0.68	0.62
Pythia	1.4B	12B	$[2,3)$	MLP	W → P (E)	8	0.75	0.66
Pythia	1.4B	12B	$[2,3)$	Linear	W → P (E)	8	0.76	0.65
Pythia	1.4B	12B	$[2,3)$	BET	W → P (E)	N/A	0.60	0.59
Pythia	1.4B	12B	$[2,3)$	MLP	W → P (SE)	-1	0.84	0.76
Pythia	1.4B	12B	$[2,3)$	Linear	W → P (SE)	-1	0.80	0.68
Pythia	1.4B	12B	$[2,3)$	MLP	W → P (SE)	8	0.80	0.71
Pythia	1.4B	12B	$[2,3)$	Linear	W → P (SE)	8	0.80	0.71
Pythia	1.4B	12B	$[2,3)$	BET	W → P (SE)	N/A	0.58	0.56

Table 7. Llama 2 binary classification results (with a gap). For training details see Section D.2, and for baseline definitions, see Section 3.4. Input activations are drawn from the last layer of the small model.

Model	S	L	Band	Type	Dataset (Train → Eval)	AUC	Acc
Llama 2	7B	70B	$[1,2)$	MLP	W → W	0.94	0.87
Llama 2	7B	70B	$[1,2)$	Linear	W → W	0.91	0.83
Llama 2	7B	70B	$[1,2)$	BET	W → W	0.57	0.55
Llama 2	7B	70B	$[1,2)$	NNE	W → W	0.72	0.67
Llama 2	7B	70B	$[2,3)$	MLP	W → W	0.93	0.84
Llama 2	7B	70B	$[2,3)$	Linear	W → W	0.90	0.83
Llama 2	7B	70B	$[2,3)$	BET	W → W	0.55	0.55
Llama 2	7B	70B	$[2,3)$	NNE	W → W	0.74	0.64
Llama 2	7B	70B	$[3,4)$	MLP	W → W	0.90	0.80
Llama 2	7B	70B	$[3,4)$	Linear	W → W	0.87	0.79
Llama 2	7B	70B	$[3,4)$	BET	W → W	0.54	0.54
Llama 2	7B	70B	$[3,4)$	NNE	W → W	0.70	0.65
Llama 2	7B (chat)	70B	$[2,3)$	MLP	W → W	0.90	0.84
Llama 2	7B (chat)	70B	$[2,3)$	Linear	W → W	0.90	0.83
Llama 2	7B (chat)	70B	$[2,3)$	BET	W → W	0.49	0.53

"small" model allows more obvious "epistemic" tokens to pass the evaluation set filter. However, classifiers were more unstable for the smallest revision, for which the MLP repeatedly failed to converge. Also, classifiers trained on the oldest revisions do not transfer as well to new distributions.

F.8. Binary classification full results (without gap)

Results for the experiments described in Section 4.2 are provided in Table 10. While lower than the results on the gapped task, mostly on account of previously absent points near the decision threshold, the classifiers are still accurate, with AUC scores > 0.8 across the board compared to baseline scores < 0.55. The choice of threshold has a surprisingly small effect

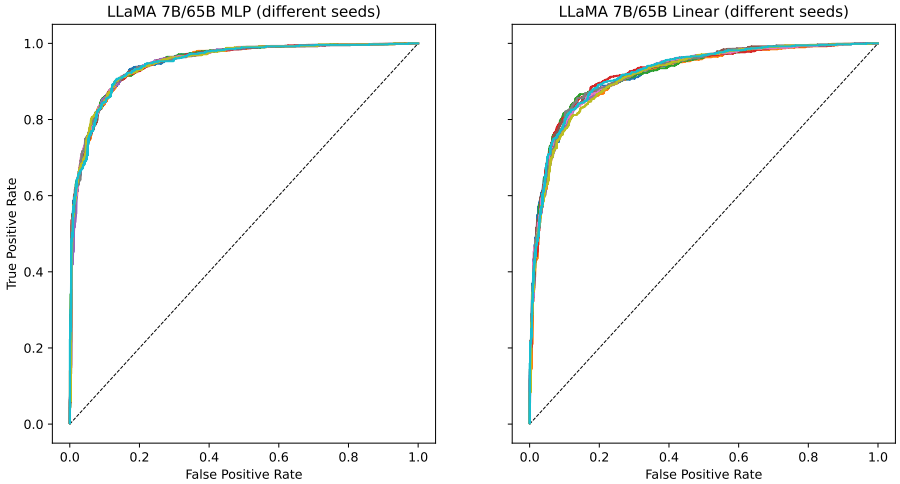

Figure 11. **Classifier performance is not sensitive to training randomness.** ROC curves for classifiers trained on the Wikipedia set using LLaMA 7B and LLaMA 65B for the binary classification task (with gap). Runs (10 per pane) differ only by the choice of random seed.

Table 8. Binary classification results (with a gap) on an *unseen model pairing*. Classifiers are trained using one large model and evaluated using labels from another. Input activations are drawn from the last layer of the small model.

MODEL	S	L (TRAIN → EVAL)	BAND	TYPE	DATASET (TRAIN → EVAL)	AUC	ACC
LLAMA	7B	30B → 65B	$[2, 3)$	MLP	W → W	0.89	0.82
LLAMA	7B	30B → 65B	$[2, 3)$	LINEAR	W → W	0.87	0.80
LLAMA	7B	30B → 65B	$[2, 3)$	MLP	W → P	0.80	0.72
LLAMA	7B	30B → 65B	$[2, 3)$	LINEAR	W → P	0.76	0.69
LLAMA	7B	30B → 65B	$[2, 3)$	MLP	W → P (C)	0.77	0.70
LLAMA	7B	30B → 65B	$[2, 3)$	LINEAR	W → P (C)	0.74	0.68
LLAMA	7B	30B → 65B	$[2, 3)$	MLP	W → P (E)	0.71	0.65
LLAMA	7B	30B → 65B	$[2, 3)$	LINEAR	W → P (E)	0.66	0.61
LLAMA	7B	30B → 65B	$[2, 3)$	MLP	W → P (SE)	0.84	0.76
LLAMA	7B	30B → 65B	$[2, 3)$	LINEAR	W → P (SE)	0.82	0.74

on the performance of the final classifiers.

To quantify the performance of these classifiers near their respective decision boundaries, we include a histogram of the distances between target values entropy values and the threshold (in bits) for both misclassified and classified points, using the 7B/65B MLP in the band [2, 3) with a threshold of 1, in Figure 13. As expected, accuracy near the boundary is essentially 50% and higher farther away from it. Evaluating the same classifier using a filter including a gap with standard hyperparameter settings (i.e. a filter that excludes all points for which the large model's entropy is in the range [0.2, 2) and then rebalances the classes accordingly) yields an AUC score of 0.89 (up from 0.83) and an accuracy of 0.84 (up from 0.76), both comparable to the scores of our "gapped" classifiers.

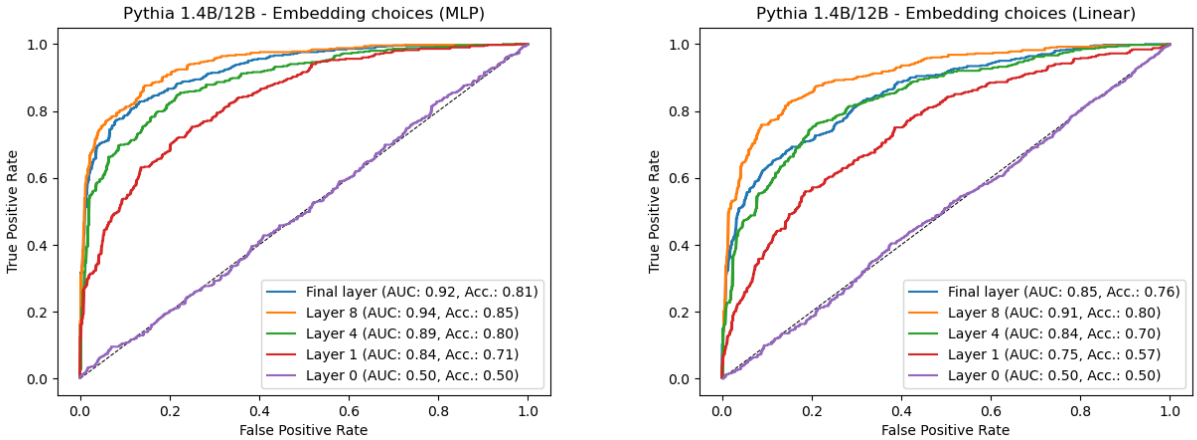

Figure 12. Classification experiments (with a gap) using activations from *different layers* in the small model. *Left:* nonlinear classification. *Right:* linear classification. We use Pythia 1.4B as the small model and Pythia 12B as the large model. Classification heads are trained and tested on Wikipedia data with entropy band $[2.0, 3.0)$.

Table 9. Binary classification results (without a gap) on the Wikipedia test set using early checkpoints of the "small" model. "Step" denotes the Pythia training step of the small model checkpoint. Input activations are drawn from the last layer of the small model.

MODEL	S	STEP	L	BAND	TYPE	DATASET (TRAIN → EVAL)	AUC	ACC
PYTHIA	1.4B	9000	12B	$[2,3)$	MLP	W → W	0.54	0.51
PYTHIA	1.4B	9000	12B	$[2,3)$	LINEAR	W → W	0.91	0.84
PYTHIA	1.4B	9000	12B	$[2,3)$	BET	W → W	0.46	0.66
PYTHIA	1.4B	9000	12B	$[2,3)$	LINEAR	W → P (C)	0.57	0.56
PYTHIA	1.4B	9000	12B	$[2,3)$	LINEAR	W → P (E)	0.68	0.64
PYTHIA	1.4B	9000	12B	$[2,3)$	LINEAR	W → P (SE)	0.69	0.66
PYTHIA	1.4B	18000	12B	$[2,3)$	MLP	W → W	0.92	0.84
PYTHIA	1.4B	18000	12B	$[2,3)$	LINEAR	W → W	0.89	0.80
PYTHIA	1.4B	18000	12B	$[2,3)$	BET	W → W	0.46	0.50
PYTHIA	1.4B	35000	12B	$[2,3)$	MLP	W → W	0.92	0.81
PYTHIA	1.4B	35000	12B	$[2,3)$	LINEAR	W → W	0.86	0.76
PYTHIA	1.4B	35000	12B	$[2,3)$	BET	W → W	0.48	0.50
PYTHIA	1.4B	35000	12B	$[2,3)$	MLP	W → W	0.76	0.81
PYTHIA	1.4B	35000	12B	$[2,3)$	LINEAR	W → W	0.76	0.69
PYTHIA	1.4B	35000	12B	$[2,3)$	BET	W → W	0.48	0.50
PYTHIA	1.4B	70000	12B	$[2,3)$	MLP	W → W	0.87	0.78
PYTHIA	1.4B	70000	12B	$[2,3)$	LINEAR	W → W	0.76	0.69
PYTHIA	1.4B	70000	12B	$[2,3)$	BET	W → W	0.54	0.53
PYTHIA	1.4B	143000	12B	$[2,3)$	MLP	W → W	0.91	0.82
PYTHIA	1.4B	143000	12B	$[2,3)$	LINEAR	W → W	0.86	0.79
PYTHIA	1.4B	143000	12B	$[2,3)$	BET	W → W	0.59	0.59

Table 10. Binary classification results (without a gap) on the Wikipedia test set (after balancing). "Threshold" denotes the boundary between the two bins (in bits). "Layer" denotes the layer of the small model (out of 32) from which classifier inputs are drawn.

MODEL	S	L	BAND	TYPE	THRESHOLD	LAYER	AUC	ACC
LLAMA	7B	65B	$[2, 3)$	MLP	1	-1	0.83	0.76
LLAMA	7B	65B	$[2, 3)$	LINEAR	1	-1	0.80	0.74
LLAMA	7B	65B	$[2, 3)$	MLP	1	16	0.85	0.77
LLAMA	7B	65B	$[2, 3)$	LINEAR	1	16	0.83	0.75
LLAMA	7B	65B	$[2, 3)$	BET	1	N/A	0.58	0.56
LLAMA	7B	65B	$[2, 3)$	MLP	0.5	-1	0.84	0.77
LLAMA	7B	65B	$[2, 3)$	LINEAR	0.5	-1	0.83	0.75
LLAMA	7B	65B	$[2, 3)$	BET	0.5	N/A	0.55	0.54
LLAMA	7B	65B	$[2, 3)$	MLP	0.2	-1	0.84	0.77
LLAMA	7B	65B	$[2, 3)$	LINEAR	0.2	-1	0.82	0.76
LLAMA	7B	65B	$[2, 3)$	BET	0.2	N/A	0.54	0.54
LLAMA	7B	65B	$[3, 4)$	MLP	1	-1	0.82	0.75
LLAMA	7B	65B	$[3, 4)$	LINEAR	1	-1	0.81	0.73
LLAMA	7B	65B	$[3, 4)$	BET	1	N/A	0.55	0.55
LLAMA	7B	65B	$[3, 4)$	MLP	0.5	-1	0.81	0.73
LLAMA	7B	65B	$[3, 4)$	LINEAR	0.5	-1	0.80	0.73
LLAMA	7B	65B	$[3, 4)$	BET	0.5	N/A	0.52	0.53
LLAMA	7B	65B	$[3, 4)$	MLP	0.2	-1	0.83	0.75
LLAMA	7B	65B	$[3, 4)$	LINEAR	0.2	-1	0.80	0.73
LLAMA	7B	65B	$[3, 4)$	BET	0.2	N/A	0.52	0.52

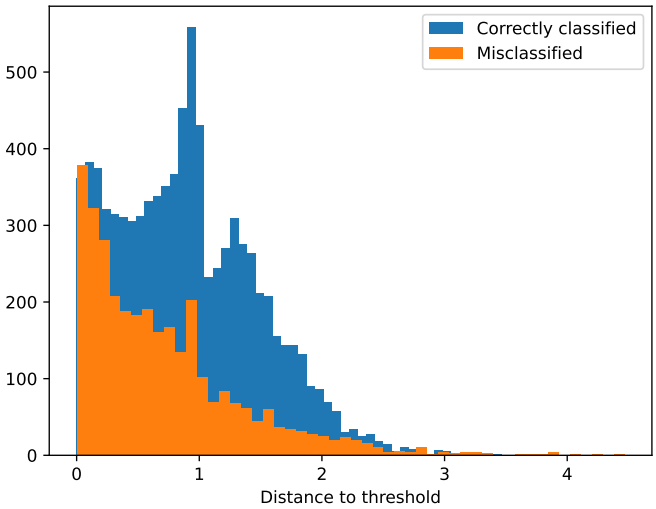

Figure 13. The absolute difference between ground truth large model predictive entropy targets and the classification boundary (1 bit) for examples correctly classified and misclassified by a 7B/65B binary classifier for Wikipedia tokens in the small-model entropy band [2, 3). Near the boundary, accuracy is approximately 50%, as expected. Accuracy far from the boundary approaches that of classifiers trained with an artificial gap in the distribution of ground truth values.

G. Regressions

Because thresholds for "near-zero" entropy vary from domain to domain, accurate regressions for our tasks would be more immediately useful than simple binary classifiers. In this section, we experiment therewith.

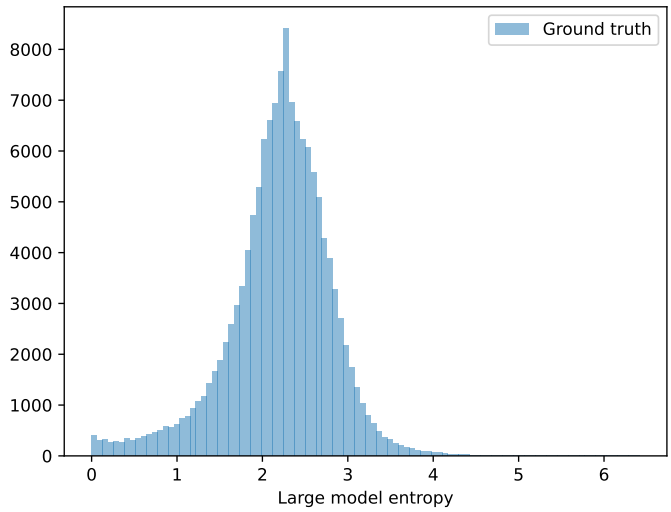

Figure 14. Distribution of target values in the small model entropy band [2,3) of the Wikipedia validation set. Most values are clustered within the small entropy band, and few are significantly smaller.

An immediate challenge is that the distribution of target values is extremely imbalanced (see Figure 14), even within 1-bit small-model entropy bands, while at the same time we care the most about examples for which the large model has small entropy conditioned on the small model being uncertain (for the same reasons we discuss in the main paper, *e.g.* that we assume the small model is decently well-calibrated and is rarely "delusionally confident"). We attempt the following interventions to force our regressions to learn something akin to what our classifiers do:

- **Upsampling (U):** Somewhat similarly to (Yang et al., 2021), within a small-model entropy band, we upsample the (rare) tokens for which the target value is significantly smaller than the mean. Concretely, we set the probability of accepting some uniformly sampled prompt x to

$$P(x) = \min\left\{1, \frac{1}{\max\{\epsilon, \alpha * H(L(t))\}}\right\}$$

  where H is the entropy function, L is the label-generating large model, as before, and α and ϵ are tunable constants that depend on the degree of imbalance in the training distribution.

- **Punishing underestimates (PU):** Of course, upsampling low-frequency tokens with small target values risks simply shifting the distribution of predictions on imbalanced data left, drowning useful signal. To improve "precision" for tokens with a small target value, we add a term to the loss to punish underestimates of the target value. For predictions x and target values y, we compute the loss as follows:

$$\mathcal{L}(x, y) = \underbrace{(x-y)^2}_{\text{squared error}} + \alpha \underbrace{(\max\{y-x, 0\})^2}_{\text{squared underestimate}}$$

  where α is again a tunable constant.

Results for Wikipedia are given in Table 11. Clearly, while results are nontrivial, more work is needed to make the regressions usable in the imbalanced case.

Table 11. Regressions trained on Wikipedia data and evaluated on the Wikipedia test set. MSE is standard mean squared error on the test set. SME is the "small model entropy" baseline. We evaluate the performance of the regressions as binary classifiers with various thresholds at inference time.

MODEL	S	L	BAND	TYPE	MSE	THRESHOLD	PRECISION	RECALL
LLAMA	7B	65B	$[2, 3)$	MLP	0.29	1	0.70	0.14
LLAMA	7B	65B	$[2, 3)$	MLP	0.29	0.5	0.61	0.08
LLAMA	7B	65B	$[2, 3)$	MLP	0.29	0.2	0.46	0.06
LLAMA	7B	65B	$[2, 3)$	LINEAR	0.37	1	0.69	0.02
LLAMA	7B	65B	$[2, 3)$	LINEAR	0.37	0.5	0.75	0.001
LLAMA	7B	65B	$[2, 3)$	LINEAR	0.37	0.5	N/A	0
LLAMA	7B	65B	$[2, 3)$	MLP (+ PU)	0.33	1	0.74	0.11
LLAMA	7B	65B	$[2, 3)$	MLP (+ PU)	0.33	0.5	0.71	0.06
LLAMA	7B	65B	$[2, 3)$	MLP (+ PU)	0.33	0.2	0.61	0.05
LLAMA	7B	65B	$[2, 3)$	LINEAR (+ PU)	0.40	1	0.48	0.01
LLAMA	7B	65B	$[2, 3)$	LINEAR (+ PU)	0.40	0.5	0.23	0.003
LLAMA	7B	65B	$[2, 3)$	LINEAR (+ PU)	0.40	0.5	0.29	0.002
LLAMA	7B	65B	$[2, 3)$	MLP (+ U, PU)	0.41	1	0.49	0.27
LLAMA	7B	65B	$[2, 3)$	MLP (+ U, PU)	0.41	0.5	0.57	0.08
LLAMA	7B	65B	$[2, 3)$	MLP (+ U, PU)	0.41	0.2	0.65	0.03
LLAMA	7B	65B	$[2, 3)$	LINEAR (+ U, PU)	0.51	1	0.38	0.26
LLAMA	7B	65B	$[2, 3)$	LINEAR (+ U, PU)	0.51	0.5	0.31	0.07
LLAMA	7B	65B	$[2, 3)$	LINEAR (+ U, PU)	0.51	0.5	0.28	0.03
LLAMA	7B	65B	$[2, 3)$	SME	0.39	-	-	-

H. Unsupervised classification (extended)

H.1. Additional Results for ICLT Method on LLaMA

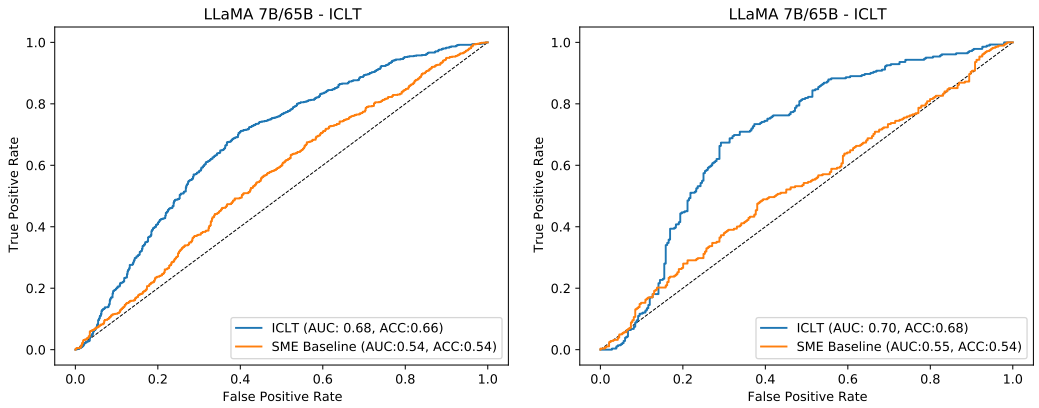

Figure 15. ROC curves for ICLT method compared to SME baseline. *Left:* Entropy bin $[2-3)$. *Right:* Entropy bin: $[3-4)$

Table 12. Llama2 ICLT results. "Band" denotes lower and upper bounds of small model predictive entropy.

S	L	BAND	BASELINE		REPETITION	
			AUC	ACC	AUC	ACC
7B	70B	$[1.0, 2.0)$	0.57	0.55	0.65	0.61
7B	70B	$[2.0, 3.0)$	0.54	0.57	0.68	0.63
7B	70B	$[3.0, 4.0)$	0.56	0.55	0.70	0.64

In Table 12 we show ICLT results on Llama 2. In Figure 16 and Figure 17 we also show additional examples where the ICLT works less well. By manually inspecting examples, we have some preliminary understandings on reasons behind why ICLT method does not work. First, our tagging for aleatoric or epistemic labels comes from large model - in other words, we are applying ICLT method to complete the same classification task as specified in Section 4.1. However, there are instances where humans may disagree with the large model on whether the example is epistemic or aleatoric. As mentioned, the large model can exhibit epistemic uncertainty and thus provides imperfect labels. Example C in Figure 16 is one where the label provided by the large model is aleatoric but humans are likely to consider the case to be epistemic. On the other hand, we also observe there are cases where large model is fairly certain about cases where humans might consider the case to be aleatoric - Example A, B in Figure 17. In Appendix A, we further discuss how large mode tagging could affect the dataset and hence the nature of our classfication task. Second, the aleatoric or epistemic nature may vary on a token-specific level. For example, we have noticed several cases, where the prompt ends with an numerical value and if the next token prediction is an additional numerical digit to complete the numerical value provided in the prompt, we may consider the example to be epistemic. However, if the model thinks the value in the prompt is complete, then the prompt may be more aleatoric in nature. In those cases, we observe that the model behaves very differently based on the context provided in our ICLT context (See Example B in Figure 16. Finally, the last category includes examples where examples appear less clear to humans whether the nature of the uncertainty, where there seems to be a mix of both aleatoric and epistemic. However, by the setup of our classification task, the large model is either super confident or less confident in those cases (See Example C in Figure 16 and Example C in Figure17), and we use large model as the ground truth label.

Below is the prompt used for the six examples given in Figure 16 and Figure 17.

- **Figure 16:** Examples where large model is uncertain about next token prediction (aleatoric)
  - **Example A** "Australian Irish Sign Language or AISL is a minority sign language in Australia. As a Francosign language, it is related to French Sign Language as opposed to Auslan which is a Banzsl language which is related

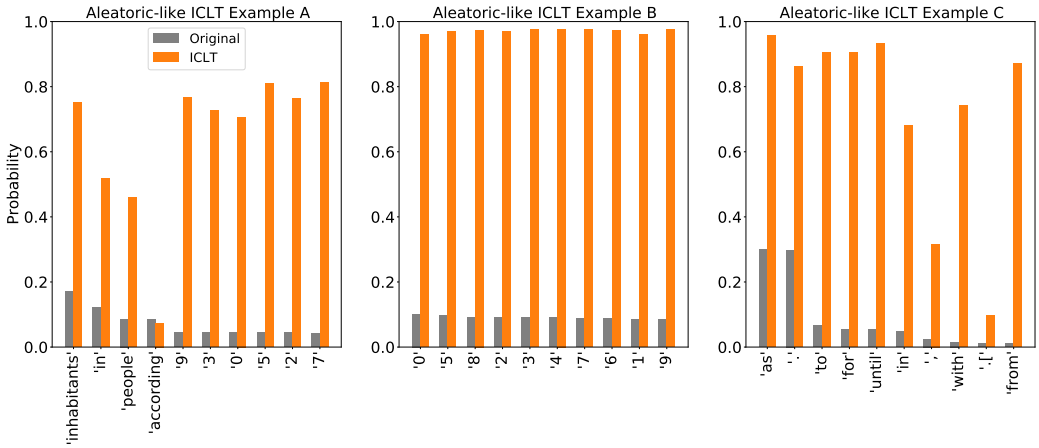

Figure 16. Additional examples for the ICLT method using LLaMA 7B model to predict LLaMA 30B model's entropy. In these examples, LLaMA 30B is unsure about the next token, hence the uncertainty is aleatoric but the ICLT fails at classifying examples to be aleatoric.

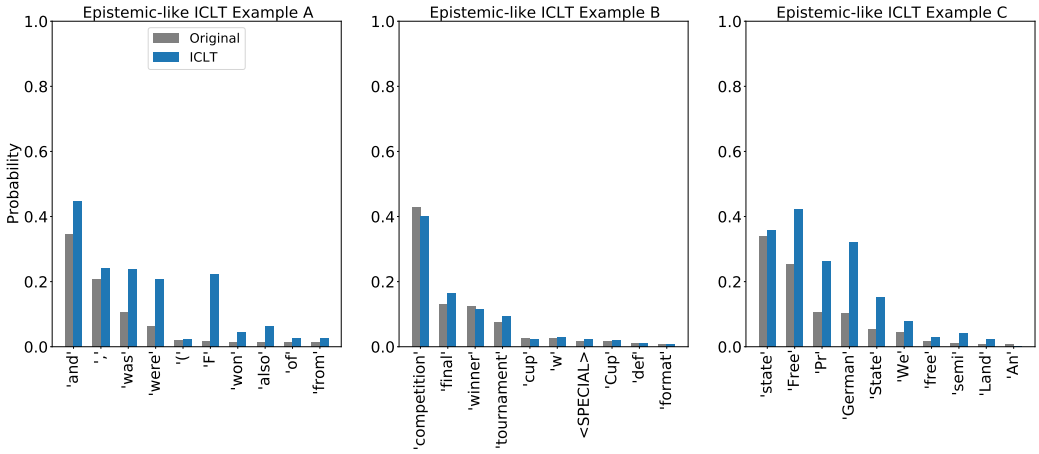

Figure 17. Additional examples for the ICLT method using LLaMA 7B model to predict LLaMA 30B model's entropy. In these examples, LLaMA 30B is sure about the next token, making the uncertainty epistemic, but ICLT fails at classifying examples as such.

to British Sign Language. AISL was brought to Australia from Ireland in 1875 by a group of Dominican nuns (including a Deaf nun) where three schools were established and used AISL"

  - **Example B** "Al Mudhafr Al-Asfal is a sub-district located in the Al Bayda District, Al Bayda Governorate, Yemen. Al Mudhafr Al-Asfal had a population of 3527"
  - **Example C**"Anastas Ngjela (3 March 1934 – 2"

- **Figure 17:** Examples where large model is certain about next token prediction (epistemic)
  - **Example A** "The 1987–88 Gamma Ethniki was the fifth season since the official establishment of the third tier of Greek football in 1983. Atromitos and Makedonikos were crowned champions in 'Southern' and 'Northern' Group respectively, thus winning promotion to Beta Ethniki. Rethymniak"
  - **Example B** "The 1921–22 City Cup was the 24th edition of the City Cup, a cup competition in Northern Irish football. The"
  - **Example C** "The June 1924 Anhalt state election was held on 22 June 1924 to elect the 36 members of the Landtag of the"

H.2. Ablation study on context provided for ICLT

To further understand the ICLT method, we conducted ablation studies on different contexts provided with full results shown in table 13. We discovered that in general, the ICLT method is not super sensitive to the information provided in the context. In particular, we experimented the following changes to the context provided.

Additional Context We provided additional context by allowing the small model to auto-repressively generate next token until it outputs an period token or an $\langle \texttt{EOS} \rangle$ token to indicate the end of sentence. Then we prepended the completed sentence as the context information before the original prompt, and feed back into the model again for next-token generation.

Irrelevant Context We provided irrelevant context after the relevant context and before we repeat the promot:

$$\langle \texttt{BOS} \rangle \texttt{prompt} + \texttt{relevant context} + \texttt{irrelevant information} + \langle \texttt{BOS} \rangle \texttt{prompt} \tag{1}$$

None top k Instead of sampling top k tokens as to generate the context, we used random tokens to generate the context.

Table 13. ICLT Method ablation on different kinds of context provided. Overall, the method is not super sensititive to the context provided. LLaMA 7B on wikipedia dataset with entropy band $[2.0, 3.0)$

Ablation Type	AUC	Acc
Original ICLT	0.68	0.63
Additional context	0.70	0.65
Irrelevant context	0.64	0.62
Top 1	0.65	0.61
Top 5	0.67	0.63
Top 20	0.68	0.64
Random 10	0.61	0.60

Metric choice The choice of using minimum entropy as our metric choice comes from the ablation study above. By using the top k generations, one might want to consider using the original softmax probability as a weighting scheme. Some examples might include: weighted entropy from the ICLT generation or mutual information between the original and the ICLT generation. Because the ablation on context information suggests that the ICLT method is not sensitive to the specific kind of context provided, a natural metric choice for the unsupervised task is to exclude any information derived from the context. Furthermore, from the synthetic set-up to examining examples, our understanding is that we are looking for the copying (*i.e.* learning) behaviors from the context as an indicator for epistemic uncertainty. Therefore, we used the minimum entropy among all contexts as a metric.

H.3. Why does ICLT fail on Pythia?

Pythia has the tendency to repeat everything in verbatim from the context regardless of the prompt, as shown in Figure 18. Furthermore, such behavior exists both on Wikipedia dataset and Pile test set (shown in Table 14), and exists on all Pythia model sizes. While we did not reach a clear answer on the cause, we discovered that the document separation tokens (i.e., special tokens such as $\langle \texttt{EOS} \rangle$ and $\langle \texttt{BOS} \rangle$) plays an important role in the success in ICLT method on LLaMA, shown in Table 3.LLaMA uses a bytepair encoding (BPE) tokenizer that has $\langle \texttt{EOS} \rangle$ and $\langle \texttt{BOS} \rangle$ as two separate tokens indicating the beginning and end of the document. On the other hand, Pythia uses a BPE tokenizer (Biderman et al., 2023) specifically trained on Pile, and only end of text is indicated. In our original setup, we use $\langle \texttt{BOS} \rangle$ to indicate the document separation boundary. Inserting additional $\langle \texttt{EOS} \rangle$ token does not affect the performance for LLaMA but slightly improves the performance for Llama2. Replacing the $\langle \texttt{BOS} \rangle$ with an $\langle \texttt{EOS} \rangle$ token in between the ICLT context significantly affects the model's behavior. Finally, removing any document boundaries ("None" in Table 3) leads to ICLT failing completely.

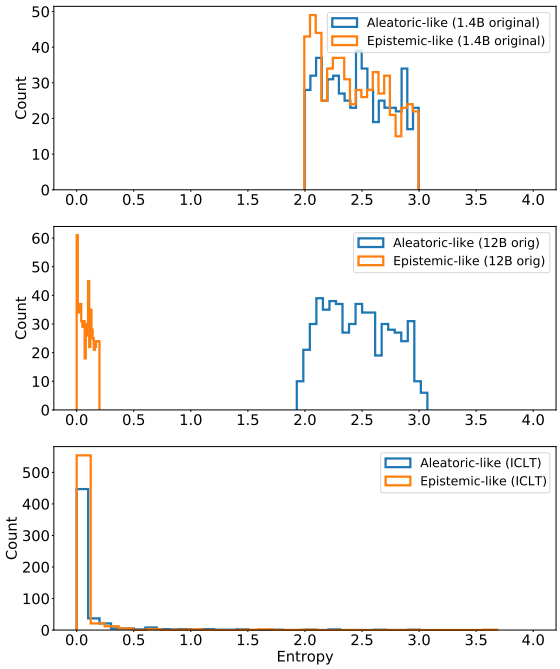

Figure 18. ICLT method on Pythia 1.4B using Pythia 12B to generate labels with Pile test set. *Top:* original entropy prediction by 1.4B model. *Middle:* entropy tagging with 12B model. *Bottom:* Minimum entropy from ICLT method on 1B.

Table 14. Unsupervised ICLT method does not work on Pyhtia: using small models (Pythia 70M, 410M, and 1.4B) to classify when large model (Pythia 12B) is confident. Entropy band $[2.0, 3.0)$. Dataset: Pile validation set. Baseline: BET

S	L	COUNT	BASELINE		REPETITION	
			AUC	ACC	AUC	ACC
70M	12B	1574	0.51	0.56	0.58	0.51
410M	12B	1042	0.59	0.55	0.61	0.56
1.4B	12B	1250	0.57	0.53	0.54	0.55

I. Labeled examples

Below, we provide labeled sentences for the 7B/65B LLaMA pairing in the [2, 3) entropy band. Individual tokens are separated by spaces. Tokens in the "epistemic" category are colored green. Red tokens are in the second class ("aleatoric-like"). *Note that labels have been class- and token- balanced and that only tokens with small model entropy between 2 and 3 are eligible to be colored here in the first place; green tokens are a very sparse subset of all "epistemic" tokens that can be detected by contrasting model entropies.*. Examples are individual sentences containing labels from randomly selected documents in the Wikipedia and Pile (Code) datasets.

Wikipedia:

- Č T 3 (Č esk á te lev ize 3 , " T ro j ka ") was the Czech public television channel , operated by Czech Television . Č T 3 broadcast originally in 1 9 9 3 when it replaced the previous station OK 3 . Un like the other two channels of the Czech Television at the time , Č T 1 and Č T 2 , Č T 3 broadcast its program largely in foreign languages and 2 4 hours a day .
- Q ar hd and al - F ard at () is a sub - d istrict located in Al Ash ah District , ' Am ran Governor ate , Y emen . Q ar hd and al - F ard at had a population of 2 6 2 3 according to the 2 0 0 4 census .
- Qu id Pro Qu o is an 1 8 4 4 comedy play by the British writer Catherine G ore , best known for her nov els . It premier ed at the Theatre Royal , Hay market in London on 1 8 June 1 8 4 4 . The original cast included Louis a C ran st oun N is b ett as Lord Bell am ont , Robert Str ick land as Jer emy Gr ig son , John Buck stone as Captain Si ppet , William Far ren as Sir George M ord ent , Henry How e as R ivers , Julia Ben nett as Lady Mary R ivers , Julia Glo ver as Mrs . Gr ig son , Mrs . Ed win Y arn old as Ellen and Anne Hum by as Br idget Prim .
- This ere ct , rh iz om at ous shr ub grows up to tall . Ro ots grow from trailing branches and many short sho ots . The branches are rig id and have a diameter of up to . Second ary branches develop on the leaf ax ils on the main stem and have a diameter of up to . Bra chy bl asts (sh o ots) grow in the leaf ax ils of the secondary ban ches . These typically grow up to long and secondary bra chy bl asts are rare . They are white when young .

  ## Le aves

  The tri angular leaves grow closely against the branches and are w ool ly on the upper surface . They are bright green and are slightly in rolled . The leaves growing on the secondary branches are about half the size of those growing on the main st ems .
- Arg im und was a Vis ig oth ic us ur per who briefly claimed the king ship in 5 8 9 – 5 9 0 before being put down by the legit imate so ver eign , Re cc ared I . Following Re cc ared ' s conversion from A rian ism to Catholic ism , a consp i racy , led by Sun na , the A rian bishop of M ér ida , arose to at first place the A rian Seg ga on the throne but failed due to the plot being bet rayed by one of its own named W itter ic . But more u pr is ings followed & amp ; this one was no different as Arg im und revol ted in 5 8 9 , somewhere in the Kingdom . In response , Re cc ared like with the re bell ion of Seg ga a few years back , sent his General Claud ius to put it down . The Revol t probably only last ed a short few months if not weeks , and Arg im und was captured . Arg im und probably had his hands cut off (like his prede cess or) and was ban ished into Ex ile , his further fate afterwards are unknown .

Code:

- ⟨ height ⟩ 2 9 5 ⟨/ height ⟩

  ⟨/ rect ⟩

  ⟨/ property ⟩

  ⟨ widget class =" Q Tab Widget " name =" tab Widget "⟩
- 2 4 3 1 2 1 7 0 2 0 2 1 6 5 1 6 4

  2 5 3 1 1 3 7 0 2 0 2 1 6 6 1 6 5

  2 6 3 7 9 0 0 2 0 2 1 6 7 1 6 6

  2 7 3 1 1 5 2 0 2 0 2 1 6 8 1 6 7

2 8 3 1 2 3 7 0 2 0 2 1 6 9 1 6 8

2 9 3 2 7 2 0 2 0 2 1 7 0 1 6 9

3 0 3 6 0 0 0 2 0 2 1 7 1 1 7 0

3 1 3 1 2 3 0 0 2 0 2 1 7 2 1 7 1

3 2 3 1 3 2 4 0 2 0 2 1 7 3 1 7 2

3 3 3 7 3 6 0 2 0 2 1 7 4 1 7 3

3 4 3 4 6 3 0 2 0 2 1 7 5 1 7 4

3 5 3 1 2 3 2 0 2 0 2 1 7 6 1 7 5

3 6 3 1 4 1 1 0 2 0 2 1 7 7 1 7 6

3 7 3 2 5 9 0 2 0 2 1 7 8 1 7 7

- fragment : string

  Fragment sh ader code

  color : string ' local ', ' shared ' or ' global ' """

  base _ d type = [(' position ', (np . float 3 2 , 3), ' ! local ', (0 , 0 , 0)),

  (' id ', (np . float 3 2 , 1), ' ! local ', 0),

  (' color ', (np . float 3 2 , 4), ' local ', (0 , 0 , 0 , 1)),

  (" linewidth ", (np . float 3 2 , 1), ' global ', 1),

  (" view port ", (np . float 3 2 , 4), ' global ', (0 , 0 , 5 1 2 , 5 1 2))

  ]

  dtype = base _ d type

  if user _ d type :

  dtype . extend (user _ d type)

  if vertex is None :

  vertex = gl sl . get (' collections / raw - path . vert ')

  if transform is None :

  transform = Null Transform ()

  self . transform = transform

  if fragment is None :

  fragment = gl sl . get (' collections / raw - path . f rag ')

- In this tutorial , we will look at how to connect an [ST M 3 2][ST M 3 2] board to the K aa platform using [K aa Io T Platform Ar duino library][K aa Io T Platform Ar duino library]. You will learn how to create a digital tw in of your device , connect it , send te lem etry and receive commands .

- bool Not ify Wait ers (const bool f Not ify All ,

  const Wait Term ination Re ason Term ination Re ason);

J. Dataset samples

To give a sense of the token distributions and formats of each of our datasets, we provide 5 random samples from each below. Wikipedia and Pile examples are drawn from the respective training sets while examples from the three earmarked Pile subsets are drawn from validation sets. We exclude Pile samples from said subsets. Some longer examples are abridged. Arrows denote linebreaks.

WIKIPEDIA

```
Topsy Chapman (1947 - 2022) was a Jazz and Gospel musician from New Orleans,
    ↪ Louisiana.

## Early life

Chapman was born in Kentwood, Louisiana. She sang and played piano at the age of
    ↪ 3 and was considered a musical prodigy. By the age of 6 or 7, she was
    ↪ earning money performing in churches.

## Career

 Chapman moved to New Orleans when she was 17 where she led her family's Gospel
     ↪ group, The Chapmans. Playwright and director, Vernel Bagneris, recruited
     ↪ the group to appear in his production on 'One Mo' Time'. The show ran at
     ↪ Toulouse Theatre and later premiered Off-Broadway. Chapman performed with
     ↪ her daughters, Yolanda Robinson and Jolynda Phillips, under the group
     ↪ name Topsy Chapman and Solid Harmony. Chapman often performed at New
     ↪ Orleans Jazz Fest and well known in the New Orleans music scene.

## Death

 Chapman died in 2022.
```

```
The Lyons Crime Family, also known as the Lyons Gang or the Lyons Clan, is a
    ↪ Scottish criminal organisation based in Glasgow. It is one of the most
    ↪ notorious criminal organisations in Scotland, with a long nistory of
    ↪ involvement in organized crime ncluding drug trafficking, extortion, and
    ↪ money laundering. The Lyons Gang is known for its rivalry with another
    ↪ Glasgow-based gang, the Daniel Crime Family (also known as the Daniel Gang
    ↪ or the Daniel Clan), which has resulted in a number of violent incidents
    ↪ over the years.

## Background

The Lyons Crime Family is believed to have been founded in the 1980s by William
    ↪ "Benny" Lyons, who was the leader of the gang at the time until his death
    ↪ in 2006. His brother, Eddie Lyons Sr is also believed to be a prominent
    ↪ member of the organisation. The family's origins can be traced back to the
    ↪ Possilpark area of Glasgow, where the Lyons family had lived for
    ↪ generations. The Lyons are known for their brutal tactics. The exact year
    ↪ when the gang was formed is not clear, but it is believed to have emerged
    ↪ in the early to mid-1980s. At that time, Glasgow was experiencing an
    ↪ increase in drug-related crime and violence, and the Lyons Crime Family
    ↪ saw an opportunity to profit from the illegal drug trade. Over time, the
```

```
    ↪ Lyons Crime Family expanded its operations beyond Glasgow and developed
    ↪ links to other criminal organisations, both in Scotland and abroad.
    ↪ Despite the arrests and convictions of some of its members, the gang has
    ↪ remained active and has continued to engage in drug trafficking,
    ↪ extortion, and other illegal activities. Over the years, the Lyons Crime
    ↪ Family has been involved in a number of high-profile crimes, including the
    ↪ murder of Kevin 'Gerbil' Carroll in 2010. The organisation has also been
    ↪ linked to a number of drug trafficking and money laundering operations.
    ↪ Scotland continues to have by far the highest drug death rate recorded by
    ↪ any country in Europe, the Lyons Gang collaborate with the notorious Irish
    ↪ Kinahan Cartel to provide high quality drugs but also to wash the money in
    ↪ the Scottish economy. Despite its criminal activities, the Lyons Crime
    ↪ Family has also been involved in charitable work in the Glasgow area,
    ↪ which has helped to increase its popularity and support among some members
    ↪ of the community. Overall, the Lyons Crime Family is a complex and
    ↪ controversial organisation that has had a significant impact on the
    ↪ criminal landscape in Scotland.

_____________________________________________________________________

Jakez Cornou (1935 - 1 September 2022) was a French historian, ethnologist, and
    ↪ author. He lived in Bigouden and pursued passions in history, ethnography,
    ↪ and heritage of the Bretons. He was a co-founder of the newspapers "" and
    ↪ "Pays de Quimper en Cornouaille". He also directed the publishing house É
    ↪ ditions Sked and was a member of the collective , which sought to promote
    ↪ literature in the "Pays Bigouden".

_____________________________________________________________________

The 2007-08 County Antrim Shield was the 119th edition of the County Antrim
    ↪ Shield, a cup competition in Northern Irish football. Glentoran won the
    ↪ tournament for the 25th time, defeating Crusaders 2-1 in the final.

_____________________________________________________________________

Cyril Aubrey Kemp (12 June 1915 - 25 December 2010) was an Irish tennis player
    ↪ active in the 1930s, 1940s and 1950s. He was also a national
    ↪ representative in the sports of squash and table tennis. The son of an
    ↪ all-round sportsman, Kemp had his best period on the tennis tour in the
    ↪ late 1940s, debuting for the Ireland Davis Cup team in 1946. He was
    ↪ singles runner-up at the Irish championships in 1946 and 1947. His run at
    ↪ the 1947 Irish championships included an upset semi-final win over Tom
    ↪ Brown, who was fresh off making a Wimbledon final. In 1948 he won through
    ↪ to the third round at Wimbledon, before losing to the top seeded Frank
    ↪ Parker.
```

PILE

```
I am so sad Charlie one of our beloved cats has cancer out Vet removed a golf
    ↪ ball sized tumor from him today . we are sending it off to be tested and i
    ↪ am praying that she got it all . We will also know what type it is but we
    ↪ fear we will have to make the decision to let him go , he is just 7 years
    ↪ old and i feel it's just too young to have to go :sad: Our other cat
    ↪ George already knows something is wrong because he keeps looking for
    ↪ Charlie. How will i help him cope if we have to put Charlie down ?
```

```
chico2

May 18th, 2006, 05:29 PM

I am sorry to hear that,I too had a cat(Peppi)who had a tennis-ball size
    ↪ tumor..I pointed it out to my vet,when the tumor was only quarter-size,she
    ↪ said it was only a fatty deposit.:mad: Peppi was also diabetic..
Weeks later it was huge and Peppi gave up on life and had to be euthanized.
My vet never suggested an operation,so that yours is attempting to remove
    ↪ Charlies could be good news.
George will eventually be ok,but if you loose Charlie:sad:George will miss him
    ↪ very much,another kitten/cat would probably help him.
But lets hope it does not come to that,lets hope Charlie recovers and that it's
    ↪ not as bad as was thought:fingerscr

                                                                    ___________

Listing 136254134. One story floor plan with high vaulted ceilings.

Many upgrades including wood flooring in all living areas, crown molding, and
    ↪ granite kitchen countertops, stainless appliances. much more. Master
    ↪ bedroom has dual sinks in the marble topped vanity and framed mirrors, and
    ↪ walk in closet and access to your private backyard by of sliding glass
    ↪ doors. Close to freeway -> Read More

                                                                    ___________

Landscapes of Communism: a History Through Buildings

Owen Hatherley

Allen Lane, 624pp, £25

When I was ten, Mum and Dad took us on our annual family holiday - this time, to
    ↪ Yugoslavia. It was, with hindsight, an especially odd place for them to
    ↪ have taken us. Mum and Dad were certainly not communists but, very much
    ↪ unlike their son, Maggie-loving Tories who'd have voted for a pig wearing
    ↪ a blue rosette. Family holidays had hitherto stuck to old favourites, such
    ↪ as Devon or Somerset. And now here we were, nonchalantly skipping across
    ↪ the Iron Curtain as if popping down to Sainsbury's. Marshal Tito's kind of
    ↪ communism was softer than most, with its open borders and political
    ↪ neutrality. But it was clear from the disappointing lack of tourist tat in
    ↪ the shops that we weren't in Lyme Regis any more, Toto. We were in heaven.

                                                                    ___________

Note: Citations are based on reference standards. However, formatting rules can
    ↪ vary widely between applications and fields of interest or study. The
    ↪ specific requirements or preferences of your reviewing publisher,
    ↪ classroom teacher, institution or organization should be applied.

Abstract:

Political theorists consider the challenge of global climate change from a range
    ↪ of perspectives, including conceptual analysis, critical theory, critical
```

```
   ↪ legal studies, and neo-Marxism.Read more...

        ____________________________________________________________

Interface

The short video to the left will show you the basic controls and operation of
   ↪ the game.

Please note: the game is in development, so there may be subtle changes between
   ↪ the video and the current state of the game

This will walk you through:

scrolling left and right (you can also scroll up and down if many annotations
   ↪ are made)

making a selection by clicking on the start and ending tokens then clicking
   ↪ annotate

deleting a selection by double clicking

clearly incorrect mentions, reserved mentions

how an agreement between two players is shown

and how to finish a round of the game

Pause and rewind the video if you miss anything.
```

PILE (CODE)

```
/* iCheck plugin Polaris skin
----------------------------------- */
.icheckbox_polaris,
.iradio_polaris {
   display: block;
   margin: 0;
   padding: 0;
   width: 29px;
   height: 29px;
   background: url(polaris.png) no-repeat;
   border: none;
   cursor: pointer;
}

.icheckbox_polaris {
   background-position: 0 0;
}
   .icheckbox_polaris.hover {
      background-position: -31px 0;
   }
   .icheckbox_polaris.checked {
      background-position: -62px 0;
```

```
   }
   .icheckbox_polaris.disabled {
      background-position: -93px 0;
      cursor: default;
   }
   .icheckbox_polaris.checked.disabled {
      background-position: -124px 0;
   }

.iradio_polaris {
   background-position: -155px 0;
}
   .iradio_polaris.hover {
      background-position: -186px 0;
   }
   .iradio_polaris.checked {
      background-position: -217px 0;
   }
   .iradio_polaris.disabled {
      background-position: -248px 0;
      cursor: default;
   }
   .iradio_polaris.checked.disabled {
      background-position: -279px 0;
   }

/* Retina support */
@media only screen and (-webkit-min-device-pixel-ratio: 1.5),
      only screen and (-moz-min-device-pixel-ratio: 1.5),
      only screen and (-o-min-device-pixel-ratio: 3/2),
      only screen and (min-device-pixel-ratio: 1.5) {
   .icheckbox_polaris,
   .iradio_polaris {
      background-image: url(polaris@2x.png);
      -webkit-background-size: 310px 31px;
      background-size: 310px 31px;
   }
}
```

```
Microsoft Visual Studio Solution File, Format Version 10.00
# Visual Studio 2008
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "Image",
   ↪ "Image\Image_vs90.vcproj", "{DA74060D-73AF-3E8F-A804-FBC960DAC393}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "Text",
   ↪ "Text\Text_vs90.vcproj", "{0DE18C25-1694-3598-831D-4FA48D113606}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "Template",
   ↪ "Template\Template_vs90.vcproj", "{27E36FB4-BDAB-3B36-910A-1F1C26853B1E}"
EndProject
```

```
/* Copyright (C) 2019 Open Information Security Foundation
 *
 * You can copy, redistribute or modify this Program under the terms of
 * the GNU General Public License version 2 as published by the Free
 * Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * version 2 along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
 * 02110-1301, USA.
 */

/**
 *
 * \author Giuseppe Longo <giuseppe@glongo.it>
 *
 * Implements the sip.stat_msg sticky buffer
 *
 */

#include "suricata-common.h"
#include "threads.h"
#include "debug.h"
#include "decode.h"
#include "detect.h"

#include "detect-parse.h"
#include "detect-engine.h"
#include "detect-engine-mpm.h"
#include "detect-engine-prefilter.h"
#include "detect-content.h"
#include "detect-pcre.h"
#include "detect-urilen.h"

#include "flow.h"
#include "flow-var.h"
#include "flow-util.h"

#include "util-debug.h"
#include "util-unittest.h"
#include "util-unittest-helper.h"
#include "util-spm.h"

#include "app-layer.h"
#include "app-layer-parser.h"

#include "detect-sip-stat-msg.h"
#include "stream-tcp.h"
```

```
#include "rust.h"
#include "app-layer-sip.h"

#define KEYWORD_NAME "sip.stat_msg"
#define KEYWORD_DOC "sip-keywords.html#sip-stat-msg"
#define BUFFER_NAME "sip.stat_msg"
#define BUFFER_DESC "sip response status message"
static int g_buffer_id = 0;

static int DetectSipStatMsgSetup(DetectEngineCtx *de_ctx, Signature *s, const
    ↪ char *str)
{
   if (DetectBufferSetActiveList(s, g_buffer_id) < 0)
      return -1;

   if (DetectSignatureSetAppProto(s, ALPROTO_SIP) < 0)
      return -1;

   return 0;
}
```

```
namespace Alphaleonis.Win32.Vss
{
  /// <summary>The <see cref="VssRestoreType"/> enumeration is used by a
      ↪ requester to indicate the type of restore operation it is about to
      ↪ perform.</summary>
  /// <remarks>
  /// <para>A requester sets the type of a restore operation using <see
      ↪ cref="IVssBackupComponents.SetRestoreState"/>.</para>
  /// <!-- <para>A writer can retrieve the type of a restore operation by
      ↪ calling CVssWriter::GetRestoreType.</para> -->
  /// </remarks>
  public enum VssRestoreType
  {
    /// <summary>
    /// <para>No restore type is defined.</para>
    /// <para>This indicates an error on the part of the requester.</para>
    /// </summary>
    Undefined = 0,

    /// <summary>The default restore type: A requester restores backed-up data
        ↪ to the original volume from a backup medium.</summary>
    ByCopy = 1,

    /// <summary>
    /// <para>
    /// A requester does not copy data from a backup medium, but imports a
        ↪ transportable shadow copy
    /// and uses this imported volume for operations such as data mining.
    /// </para>
    /// <para>
    /// <b>Windows Server 2003, Standard Edition and Windows Server 2003, Web
        ↪ Edition:</b> This value is not supported. All editions of Windows
```

```
        ↪ Server 2003 SP1 support this value.
    /// </para>
    /// </summary>
    Import = 2,

    /// <summary>A restore type not currently enumerated. This value indicates
        ↪ an application error.</summary>
    Other = 3

  };
}
```

```
// SPDX-License-Identifier: MIT
// Copyright (c) 2015-2020 Zig Contributors
// This file is part of [zig](https://ziglang.org/), which is MIT licensed.
// The MIT license requires this copyright notice to be included in all copies
// and substantial portions of the software.
// Ported from:
//
// https://github.com/llvm/llvm-project/blob/
   ↪ 2ffb1b0413efa9a24eb3c49e710e36f92e2cb50b/compiler-rt/lib/builtins/modti3.c

const udivmod = @import("udivmod.zig").udivmod;
const builtin = @import("builtin");
const compiler_rt = @import("../compiler_rt.zig");

pub fn __modti3(a: i128, b: i128) callconv(.C) i128 {
   @setRuntimeSafety(builtin.is_test);

   const s_a = a >> (128 - 1); // s = a < 0 ? -1 : 0
   const s_b = b >> (128 - 1); // s = b < 0 ? -1 : 0

   const an = (a ^ s_a) -% s_a; // negate if s == -1
   const bn = (b ^ s_b) -% s_b; // negate if s == -1

   var r: u128 = undefined;
   _ = udivmod(u128, @bitCast(u128, an), @bitCast(u128, bn), &r);
   return (@bitCast(i128, r) ^ s_a) -% s_a; // negate if s == -1
}
```

PILE (EUROPARL)

```
Tervetulotoivotukset
Puhemies
(DE) Hyvät parlamentin jäsenet, minulla on suuri ilo toivottaa tervetulleeksi
   ↪ joukko Saksan demokraattisen tasavallan ensimmäisen vapailla vaaleilla
   ↪ valitun parlamentin entisiä jäseniä, jotka istuvat yleisölehterillä.
Euroopan parlamentti pääsi historiankirjoihin päättäessään yhdistää Saksan
   ↪ uudelleen ja hajosi itse pian sen jälkeen. Valtuuskuntaa johtaa
   ↪ Volkskammerin silloinen puheenjohtaja, tohtori Sabine Bergmann-Pohl.
   ↪ Toivotan teidät erittäin lämpimästi tervetulleeksi Euroopan parlamenttiin.
(Suosionosoituksia)
```

```
4. Informações relativas a medicamentos sujeitos a receita médica (procedimentos
   ↪ comunitários de autorização e de fiscalização de medicamentos) (
Antes da votação da alteração 13:
Christofer Fjellner
Senhor Presidente, tenho uma pequena alteração oral em resultado de um
   ↪ compromisso de última hora entre os grupos políticos relativamente à
   ↪ alteração 13, cujo texto actual, que diz "no prazo de 60 dias a contar da
   ↪ data de recepção da notificação" deve ser alterado para "no prazo de 90
   ↪ dias a contar da data de recepção da notificação".
Esta alteração foi acordada entre todos os grupos políticos.
```

```
Übermittlung von Gemeinsamen Standpunkten des Rates: siehe Protokoll
Boguslaw Rogalski
(PL) Frau Präsidentin! Entschuldigen Sie bitte, aber ich möchte etwas zu meinem
   ↪ Abstimmungsverhalten bezüglich des Berichts über Mazedonien sagen, wenn
   ↪ das möglich ist.
Die Präsidentin
Herr Rogalski, Sie standen nicht auf der Liste, und ich habe die
   ↪ Stimmerklärungen für abgeschlossen erklärt. Es tut mir leid. Lassen Sie es
   ↪ uns das nächste Mal bitte innerhalb der vorgeschriebenen Frist wissen. Ich
   ↪ weise Sie jedoch darauf hin, dass Sie Ihre Stimmerklärung schriftlich
   ↪ abgeben können.
(Die Sitzung wird um 12.45 Uhr unterbrochen und um 15.00 Uhr wieder aufgenommen.)
```

```
Velkomst
Formanden
Mine damer og herrer! Jeg byder velkommen til eftermiddagsmødet. Mødet er
   ↪ genoptaget og er det sidste møde før juleferien og nytår. Jeg vil gerne
   ↪ benytte lejligheden til at sige velkommen til direktionen for den tyske
   ↪ kommission for krigsgrave (Volksbund Deutsche Kriegsgräberfürsorge) fra
   ↪ Baden-Württemberg under ledelse af hr. Nothelfer. Med base i
   ↪ Baden-Württemberg står kommissionen for vedligeholdelsen af gravstederne
   ↪ for 90 000 ofre for krig og tyranni i Baden-Württemberg alene. Vi takker
   ↪ Dem for Deres arbejde og byder Dem velkommen til Europa-Parlamentet.
```

```
4. Särskild rapport från Europeiska ombudsmannen till Europaparlamentet efter
   ↪ förslaget till rekommendation till Europeiska kommissionen avseende
   ↪ klagomål 676/2008/RT (enligt artikel 205.2, 1:a delen, i arbetsordningen) (
```

PILE (STACK EXCHANGE)

```
   Q:

window.open location=yes - Firefox 7.01 not showing navigation buttons

I have a window that is created via window.open. When the window is created in
   ↪ Firefox 7.01, I turn on the address bar (location=yes). In Firefox 6.01 -
   ↪ 7.01 the forward and back buttons do not show. It works perfectly in IE.
```

```
    ↪ Can someone tell me/show me how to show navigation buttons using
    ↪ windows.open and location=yes in Firefox? Thanks!

A:

Try adding toolbar=yes as well.
location=yes is for showing the location bar.
Many browsers will default (or even force) some elements you don't request to
    ↪ show anyway, but you should be explicit about all the elements you wish to
    ↪ show rather than relying on this.
More information is available at MDN.

---

Q:

Is it ill-advised to make a cover for your thesis in STEM?

I handed in my undergraduate thesis in mathematics and since I had extra time
    ↪ due to COVID-19 lockdown, I had made a cover drawing for it that I
    ↪ included. I've never seen a thesis with a cover in the STEM field before.
Was it a bad decision on my part? Will it look like kowtowing/bootlicking my
    ↪ supervising professor for a better grade? I believe my work was pretty
    ↪ good on its own and I'm afraid I have ruined it with what was intended as
    ↪ a personal touch that I really wanted to add.

A:

It does not matter.
It used to be that theses were printed on paper. In those days, they had
    ↪ standard covers so they would all look the same when placed on a shelf in
    ↪ the library.
Today nobody cares about a digital cover.
Be prepared to provide a copy without the cover if someone asks for it.

A:

This really comes down to institutional formatting policies. Some may disallow
    ↪ cover art, some may be indifferent to it, and some actually encourage it.
    ↪ For example, my Bachelor's thesis was in a STEM field and included an
    ↪ electron microscopy image on the front cover without any issue. I
    ↪ generally think it's a nice touch, assuming the cover art is relevant to
    ↪ the contents. Further, if that assumption holds it's really difficult to
    ↪ see how it could be interpreted as bootlicking.
I also know of several higher degree theses (Master's, PhD) from the same
    ↪ institution in physics and engineering that had front cover art. And they
    ↪ still print and bind PhD theses (and some Master's theses). They achieve a
    ↪ uniform look on a library shelf since the spine isn't affected. In fact,
    ↪ the institution's website notes that they prefer PhD theses to have
    ↪ similar cover art.

---

Q:
```

```
Intelligent spell checking

I'm using NHunspell to check a string for spelling errors like so:
var words = content.Split(' ');
string[] incorrect;
using (var spellChecker = new Hunspell(affixFile, dictionaryFile))
{
   incorrect = words.Where(x => !spellChecker.Spell(x))
      .ToArray();
}

This generally works, but it has some problems. For example, if I'm checking the
   ↪ sentence "This is a (very good) example", it will report "(very" and
   ↪ "good)" as being misspelled. Or if the string contains a time such as
   ↪ "8:30", it will report that as a misspelled word. It also has problems
   ↪ with commas, etc.
Microsoft Word is smart enough to recognize a time, fraction, or comma-delimited
   ↪ list of words. It knows when not to use an English dictionary, and it
   ↪ knows when to ignore symbols. How can I get a similar, more intelligent
   ↪ spell check in my software? Are there any libraries that provide a little
   ↪ more intelligence?
EDIT:
I don't want to force users to have Microsoft Word installed on their machine,
   ↪ so using COM interop is not an option.

A:

If your spell checker is really that stupid, you should pre-tokenize its input
   ↪ to get the words out and feed those one at a time (or as a string joined
   ↪ with spaces). I'm not familiar with C#/.NET, but in Python, you'd use a
   ↪ simple RE like \w+ for that:
>>> s = "This is a (very good) example"
>>> re.findall(r"\w+", s)
['This', 'is', 'a', 'very', 'good', 'example']

and I bet .NET has something very similar. In fact, according to the .NET docs,
   ↪ \w is supported, so you just have to find out how re.findall is called
   ↪ there.

        ____________________________________________________________

Q:

Create list of variable type

I am trying to create a list of a certain type.
I want to use the List notation but all I know is a "System.Type"
The type a have is variable. How can I create a list of a variable type?
I want something similar to this code.
public IList createListOfMyType(Type myType)
{
    return new List<myType>();
}

A:
```

```
Something like this should work.
public IList createList(Type myType)
{
   Type genericListType = typeof(List<>).MakeGenericType(myType);
   return (IList)Activator.CreateInstance(genericListType);
}

A:

You could use Reflections, here is a sample:
   Type mytype = typeof (int);

   Type listGenericType = typeof (List<>);

   Type list = listGenericType.MakeGenericType(mytype);

   ConstructorInfo ci = list.GetConstructor(new Type[] {});

   List<int> listInt = (List<int>)ci.Invoke(new object[] {});
```

```
Q:

Json data containing NaN results in error during ajax request

I an endpoint exposed via web api which spews out JSON like this:
"[{"SomeId":1,"SomeName":"Some name 1","Parameter1":1.13,"Parameter2":3.0 ...

to jquery ajax get requests. Everything works fine until the JSON contains NaN
   ↪ (not a number):
"[{"SomeId":1,"SomeName":"Some name 1","Parameter1":1.13,"Parameter2":NaN ...

If this type of data is returned the error callback is hit.
Is this a known issue? Is jquery or web api to blame? How can I mitigate this?
   ↪ Thanks.

A:

This happens because the token NaN is invalid in JSON. JSON is a subset of
   ↪ JavaScript's object initializer syntax, which doesn't include several
   ↪ aspects of it, including NaN (and Infinity, undefined, ...).
The web API providing that output is producing invalid JSON, so any conforming
   ↪ JSON parser will complain about it. You'll need to fix (or ask the
   ↪ providers of it to fix) the API.
```