ON THE MATHEMATICAL THEORY OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern transformer-based AI methods model natural language in an explicitly statistical manner, learning a distribution over texts in training, and sampling from that distribution to perform inference. We present a general mathematical theory of such statistical language models. The theory makes no explicit use of the detailed computational structure of large language models (LLMs), treating them instead as empirical representations of the distribution over texts. Seen in this light, LLMs can be recognized as stochastic processes, indexed by the natural numbers (denoting location in a text sequence) and valued in finite sets (token vocabularies). The LLM's only responsibilities are to learn an approximation to the probability measure over text sequences, and to sample from that measure. The theory thus relies on the structure of natural language distributions to explain properties and behaviors of LLMs. We introduce the idea of *natural language as a statistical mixture model*, wherein documents pertaining to a specialized topic are sampled from a *dialect* distribution specific to that topic, and the distribution of the full training corpus is a mixture model whose components are the dialect distributions, weighted proportionally to the prevalence of their samples in the corpus. We analyze the consequences of this assumed structure for model prompting, and for fine-tuning. We confirm the theory's predictions by training a small `bert` model on samples generated from a small model language with a known distribution. We also check predictions with natural language, using the Library of Congress Classification (LCC) to supply dialect labels.

1 INTRODUCTION

The introduction of the transformer architecture (Vaswani et al., 2017) inaugurated a new epoch in the history of natural language processing (NLP), and marked the advent of the modern large language models (LLMs) that are the mainstays of current AI. With the rapid development and spread of LLMs into science and society, the need for robustly explainable AI (XAI) has grown more urgent, as undesirable AI output (AKA "AI hallucinations") is becoming ever more consequential and in need of control(Khairallah, 2026; Dai & Taylor, 2025). One way of casting the difficulty of XAI is to note that LLMs, in common with all deep learning methods, are essentially heuristic methods, not being underpinned by any well-developed mathematical theory capable of making strong predictions concerning their behavior.

Efforts to supply mathematical frameworks encompassing deep learning (DL) include Geometric Deep Learning (GDL, Bronstein et al. (2021)) and Categorical Machine Learning (CML, Shiebler et al. (2021)). By and large, these have been developed in the context of geometric or geometrizable data. GDL, in particular, only makes sense in the context of inputs and outputs with equivariant symmetries. The trouble is that NLP data is, in its native state, not at all geometric, consisting instead of sequences of tokens from finite vocabularies. These are usually embedded in geometrized Euclidean spaces for the purpose of computational processing, but such spaces have arbitrary dimensions, and introduce geometric and topological structure into distributions over text that were born innocent of such structure. If one desires to reckon with NLP text distributions in their native form, geometry is probably not the best guide. Consequently, a mathematical description of NLP LLMs that enables robust XAI is still largely lacking in our view.

In our attempts to formulate such a description, we have found that a *model agnostic* framework is quite helpful. Model agnosticism informs a number of useful techniques in XAI (Wang, 2024), essentially by drawing a veil over the computational details of the model, focusing attention instead on how input data (which may be vectorial, sequential, network...) and output decisions (which

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

may be classifications, regressions, predictions...) are related across a black-box model, and on the stability and sensitivity of those relations. In the spirit of this model agnosticism, we focus here on distributional structure of text data and the nature of output text predictions emitted by LLMs.

The focus away from models and towards data shifts the burden of explanation of LLM behavior from the computational structure of the model to the distributional structure of text data. Here, the relevant aspect of that distributional structure is the *topical* nature of documents comprising LLM training corpuses. Consequently, the subject of topic modeling constitutes a separate thread within NLP that is relevant to the work that we present here. Topic modeling is a family of unsupervised methods for latent thematic structure discovery in a document collection from a vocabulary (Vayansky & Kumar, 2020). The core assumptions of topic modeling are: each topic is a distribution over the vocabulary, and each document is a distribution over topics. Many of these methods are essentially probabilistic and generative, and many bear a family relationship or at least some intellectual debt to Latent Dirichlet Allocation (LDA) (Blei et al., 2003). A related work that bears some resemblance to our approach and to topic modeling is "in-context learning" (Xie et al., 2021), wherein Bayesian conditioning on prompts in the presence of a latent "concept" index is used to model LLM output consequent on prompts not present in the training data, through inference of common concepts.

LLMs are not topic models. Nonetheless, we will show that the topical structure of text training corpora *induces* latent topical structure in LLM output, and that explanations and interpretation of properties of LLM inference become quite straightforward in the presence of the assumption of topic structure induced by training. In particular, we will endeavor to supply explanations of two phenomena of LLM inference: the nature of effective LLM prompting, and the nature and limits of LLM fine-tuning.

1.1 Our Contributions

The novelty of our contributions in this article is as follows: (1) We point out the explicit connection between LLM inference and stochastic process theory, with the LLM learning the process' probability measure in training and supplying it in inference; (2) We introduce the idea of the distribution of natural language corpora as *statistical mixtures*, with mixture component distributions representing *dialects* of a common language that correspond to a particular topic. A natural language training corpus is composed of documents sampled from the dialect distributions, with weights corresponding to the prevalence of documents from the correponding topic. (3) We elaborate the consequences of the mixture-of-dialects structure of training data distributions for LLM prompting and LLM fine-tuning. (4) We introduce topic classification using the Library of Congress Classification (LCC) as topic labels, and classification of LLM-emitted texts using LCC classes to estimate dialect weights. We exhibit this methodology as a qualitative verification testbed for the theory using natural language. (5) We propose of a simple model language with topic structure, and with a known text distribution, as a quantitative testbed for the theory.

The latent variable corresponding to a single topic dialect mixture element plays a role not dissimilar to that of the "concept" latent variable in (Xie et al., 2021), and in what follows we exhibit Bayesian posterior concentration of dialect weights that is analogous to the posterior concentration over concepts in that work. An important difference is that a "dialect" is a more general construction than a "concept" in that it is meant to capture distributional aspects of extended text sequences associated with a topic, rather than functioning as an abstract latent topic label.

2 Theory

In this work, we take a strictly model-agnostic view of the operation of LLMs. This view is described in some detail in (Shilpika et al., 2026). It consists, briefly, of regarding an LLM in inference mode as a stochastic process model, indexed in $\mathbb{Z}_0$ (the natural numbers, including 0) and valued in a token vocabulary. As shown in (Shilpika et al., 2026), the probability distribution over texts represented by the LLM can be described completely in this mathematical framework without resorting to embedding or referring to any part of the computational structure of the LLM. Conditional and unconditional probabilities of arbitrary, not-necessarily contiguous text subsequences can be reconstructed by eliciting next-token log-probabilities ("logits", in machine-learning parlance) and applying the rules of conditional probability and Bayes' rule to the results.

Our goal is to supply explanations of how prompting works to elicit cogent responses from LLMs, and to describe quantitatively the balance between fine-tuning and catastrophic forgetting. When it

comes to addressing such questions about LLM training and inference behavior, a model-agnostic outlook shifts the burden of explanation from the computational structure of the model to the distributional structure of the data. Accordingly, we make a modeling assumption about the structure of natural language distributions: such distributions are *mixture distributions*, with each mixture component pertaining to a single topic or specialized knowledge set. We refer to each such single-topic language distribution as a *dialect* of the language. The global language from which a training corpus is sampled is therefore to be viewed as a weighted mixture of such dialects.

It is easy to see that such a dialect-distribution mixture is a necessary part of the distributional description of LLM training corpuses. Consider the following two short sequences of text, one from a text on Jane Austen studies (Todd (2006), p. 36), the other from a text on compiler technology (Su & Yan (2011), p. 80):

Talking about the role of the lexical analyzer, we first should talk about the role of the compiler since the lexical analyzer is part of it. The role of the compiler is to compile or to translate a kind of languages into another, usually into a language executable on computer.

The mind in and outside the novel is inhabited by stories, including the stories implied by habitual language: the only defence against their control is constant scrutiny, not only of works but of the memory into which they have been transmuted.

A human reading these short passages would have no difficulty in classifying the first as computer science-related and the second as literary criticism. This is because despite being written in a common language, they are clearly samples from distinct distributions over text, differing in frequencies of word appearances and of word juxtapositions at various sequence lags. An LLM training corpus typically contains samples from these dialects and many others.

In principle, the labeling of topic dialects is arbitrary, and there can be greater or lesser distributional overlap among dialects. This arbitrariness of such document topic labels does not vitiate their usefulness, because if the labels have been well-chosen, they will correspond to distributional clusters, in the sense that each dialect will have lower entropy than the entire corpus.

It can be useful to think of hierarchies of sub-topics, much in the way that the Library of Congress Classification (LCC, 2021) has a hierarchical structure, so that, for example, "Finite Groups" are a sub-sub-sub dialect of "Mathematics" according to the hierarchical classification "Mathematics→Algebra→Group Theory→Finite Groups". In such cases, our model only assigns the leaf nodes of the classification tree as labels.

Many readers will be strongly reminded of topic models (Vayansky & Kumar, 2020) by this discussion. There is in fact a clear connection in the fact that we ascribe topic labels to documents, and treat those labels as latent variables. One important difference, however, is that while topic models typically ascribe multiple topics to each document, associating each document word with a topic, here we will associate a single topic with each document. Where that label is an LCC class, it will be constrained to be a leaf node of the hierarchy. Another important difference is that LLMs are *not* topic models in their computational structure. LLMs are designed to produce coherent texts, whereas topic models treat a document as a "bag of words" with no regard for syntax. Because of these differences, we find it useful to distinguish between a "topic"—a domain—and a "dialect"—a distribution over syntax-respecting texts corresponding to a topic.

We will show, assuming only that LLMs learn accurate approximations to the topic-clustered dialect mixture distribution from which their training data was sampled, that LLMs nonetheless inherit a type of topic structure from the data, despite not being topic models by design. This idea of natural language as a mixture of dialects allows straightforward explanation of LLM behaviors that are not simple to understand from inside the computational model.

Suppose that the number of dialects comprised in the distribution is D, and that dialects themselves are indexed by $d \in \{0, 1, \ldots, D-1\} \equiv \Omega_\Delta$. Suppose that tokens t are sampled from the finite set V (the token vocabulary). A document $\boldsymbol{S}$ consists of M contiguous tokens, $\boldsymbol{S} = (t_0, t_1, \ldots, t_{M-1})$, $t_k \in V$, sampled from a single dialect d. In general, we do not know the identity of d corresponding to a document $\boldsymbol{S}$ emitted at inference time by an LLM. We may, however, write for the probability of sampling $\boldsymbol{S}$

$$\Pr(\boldsymbol{S}) = \sum_{d=0}^{D-1} \Pr(\boldsymbol{S}|d)\Pr(d) \equiv \sum_{d=0}^{D-1} w_d \Pr(\boldsymbol{S}|d). \tag{1}$$

The w_d are *prior dialect weights*, learned (implicitly) by the LLM in training, while the $\Pr(\boldsymbol{S}|d)$ are the dialect distributions. The w_d values are in proportion to the prevalence of independent text samples from the corresponding dialect in the LLMs training data, and should be roughly in proportion to the number of tokens from each dialect. Observe that this linear mixture model arises directly from the law of total probability and the definition of conditional probability, given only the assumption that the overall distribution decomposes into dialect distributions.

The probability expressions in Equation (1) are somewhat in need of formalization. We defer the complete formal definition of the stochastic process probability space to §5.1 of the Appendix (see also Billingsley (1995), pp. 27-30). Here we note briefly that the sample space is the cartesian product $\Omega = \Omega_S \times \Omega_\Delta$ of the token sequence sample space Ω_S (the space of all possible infinite token sequences) and the dialect set Ω_Δ. The dialect index d plays the role of an unobserved latent variable probabilistically inferrable from an observed token sequence. Suppose $\omega \in \Omega$, $\omega = (t, d)$, $t = (t_1, t_2, \ldots) \in \Omega_S$. The random variable $T_\mu : \Omega \mapsto V$, $T_\mu(\omega) = t_\mu$ returns the μ-th token in the sequence t. A *finite* set of such variables, $\boldsymbol{T} = (T_{\mu_0}, T_{\mu_1}, \ldots, T_{\mu_{M-1}})$, together with a corresponding token assignment $\boldsymbol{t} = (t_0, t_1, \ldots, t_{M-1})$ defines a countably infinite set $\{\boldsymbol{T} = \boldsymbol{t}\} \subset \Omega_S$, whose elements are infinite token sequences having the corresponding finite token assignments. Such sets are *measurable* in the probability space, being elements of the σ-field $\mathcal{F}$ of events. Similarly, we define the random variable $\Delta : \Omega \mapsto \Omega_\Delta$, $\Delta(t, d) = d$ which returns the dialect ID of ω. Thus we may write, for example, $\{\boldsymbol{T} = \boldsymbol{t}\} \cap \{\Delta = d\}$ for the event "the subsequence at the locations $\boldsymbol{T}$ has token assignments $\boldsymbol{t}$ and the sequence was sampled from dialect d."

The LLM completes the probability space $(\Omega, \mathcal{F}, \Pr)$ by furnishing the probability measure $\Pr()$ that assigns probabilities to events in $\mathcal{F}$, which can be recovered from the next-token logits emitted at inference time by the LLM. An expression such as the summand in middle term in Equation (1) should in fact be written $\Pr(\{\boldsymbol{S} = \boldsymbol{s}\}|\{\Delta = d\}) \times \Pr(\{\Delta = d\})$. We will keep to these notational conventions hereforth.

2.1 Prompting

Let's look at what happens to the dialect weights under prompting. Suppose that we have a prompt $\boldsymbol{p}$ and a response $\boldsymbol{r}$, and that the corresponding random variables are $\boldsymbol{P}$ and $\boldsymbol{R}$, respectively. The LLM samples $\Pr(\{\boldsymbol{R} = \boldsymbol{r}\}|\{\boldsymbol{P} = \boldsymbol{p}\})$ to produce $\boldsymbol{r}$.

But,

$$\begin{aligned}\Pr\left(\{\boldsymbol{R} = \boldsymbol{r}\}|\{\boldsymbol{P} = \boldsymbol{p}\}\right) &= \sum_{d=0}^{D-1} \Pr\left(\{\boldsymbol{R} = \boldsymbol{r}\}|\{\Delta = d\} \cap \{\boldsymbol{P} = \boldsymbol{p}\}\right) \times \Pr\left(\{\Delta = d\}|\{\boldsymbol{P} = \boldsymbol{p}\}\right) \\ &= \sum_{d=0}^{D-1} \hat{w}_d(\boldsymbol{p}) \times \Pr\left(\{\boldsymbol{R} = \boldsymbol{r}\}|\{\Delta = d\} \cap \{\boldsymbol{P} = \boldsymbol{p}\}\right), \end{aligned} \tag{2}$$

where

$$\hat{w}_d(\boldsymbol{p}) \equiv \Pr\left(\{\Delta = d\}|\{\boldsymbol{P} = \boldsymbol{p}\}\right) = \frac{w_d \times \Pr\left(\{\boldsymbol{P} = \boldsymbol{p}\}|\{\Delta = d\}\right)}{\sum_{d'=0}^{D-1} w_{d'} \times \Pr\left(\{\boldsymbol{P} = \boldsymbol{p}\}|\{\Delta = d'\}\right)}. \tag{3}$$

The $\hat{w}_d(\boldsymbol{p})$ are *posterior dialect weights*. In effect, prompting re-weights the dialects in the model by a factor proportional to the per-dialect likelihood of the prompt, $\Pr(\{\boldsymbol{P} = \boldsymbol{p}\}|\{\Delta = d\})$.

This provides an explanation for how the cogency of LLM responses can grow with longer and increasingly specific prompts. For example, the prompt $\boldsymbol{p}$ ="Please write a database query to retrieve the serial numbers of all GE washing machines sold in Virginia in 1993" would have a relatively high likelihood in computer-science related dialects, a relatively low likelihood in dialects related to medicine, and a vanishing small likelihood in dialects related to 19th-Century British Literature. Adding more specific content to $\boldsymbol{p}$ improves the focusing effect of the likelihood on the posterior weights $\hat{w}_d(\boldsymbol{p})$ (e.g. "...a SQL database query," "an Oracle SQL database query", etc.), and can help focus weight on the set of specific subdialects from which the most cogent responses may be sampled.

We can similarly explain why "persona prompting" (Olea et al., 2024) (e.g. "Assume that you are a professor of computer science and an expert on database programming...") works to improve LLM output: It is not that the model semi-anthropomorphically puts on the appropriate persona: it is that

the additional terms in the prompt help increase the focusing effect of the prompt likelihood on the posterior weights.

Note an important feature of this explanation: *it requires no knowledge of the computational structure of the LLM*. All that is required is that the LLM should have learned a reasonable approximation to $\Pr()$ in training. After that, everything else follows as a consequence of probability theory and of the assumed topic structure of natural language.

2.2 Fine-Tuning and Catastrophic Forgetting

Fine-tuning a pre-trained model involves taking a small corpus of text related to a specialized task, and updating the training of the pre-trained model for a few epochs (Zhang et al., 2024). The model thus perturbed can become an "expert" at the specialized task. The reason that this process actually works is a little mysterious, because the fine-tuning corpus is quite small compared to the pre-training corpus, and the number of training epochs is small compared to the duration of pre-training, so the fact that a small perturbation of the LLM could produce such outsized gains in expertise requires an explanation.

The fine-tuning process is also *delicate*: fine-tuning for too many training epochs results in "catastrophic forgetting," a phenomenon in which the model learned in pretraining is progressively destroyed by fine-tuning (Luo et al., 2025).

We give these facts the following interpretation: fine-tuning perturbs the distribution learned in pre-training. The nature of the perturbation is twofold:

1. The model's prior weights $w_d \equiv \Pr(\{\Delta = d\})$ of the dialects d corresponding to the specialized task are boosted, while the remaining weights are depressed, to preserve the condition $\Sigma_{d=0}^{|V|-1} w_d = 1$;
2. The model's single-dialect distributions $\Pr(\{\boldsymbol{T} = \boldsymbol{t}\}|\{\Delta = d\})$ corresponding to dialects d not concerned with the specialized task are distorted from the ones learned in pre-training.

When the fine-tuning is done for a limited number of epochs, the distortion in (2) can be a small perturbation, not too damaging to the global distribution. Catastrophic forgetting occurs when these distortions become important.

At the same time, a small boost to a small number of the w_d can have a noticeable effect, because all the w_d are small to begin with (since $D \gg 1$).

So the window of opportunity for effective fine-tuning is the result of a balance: *maximize the boost to the task-specific dialect prior weights w_d, subject to a constraint limiting the damage to the learned off-topic dialect distributions*.

3 Validation Experiments

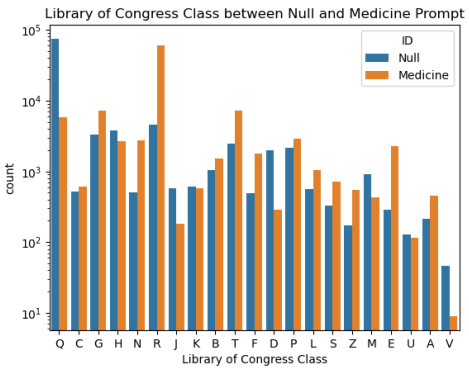
We now demonstrate that the theory of statistical language modeling just described has verifiable consequences for LLMs. We do so by means of two sets of experiments: One using a model language with known distribution to supply training texts for a small LLM, the other using models pre-trained and fine-tuned on natural language.

In both experiments, we appeal to a common set of techniques:

LLM Logit Elicitation: The response tokens emitted by LLMs are sampled according to some strategy (e.g. "greedy", top-p...) from a vector of logits computed by the final layer. When LLMs are run locally, those logit vectors are accessible and can be recorded. They are essential to recovering the probability measure $\Pr()$.

Coercive Sampling: It is frequently the case in this work that LLM output is sampled "coercively" Shilpika et al. (2026). This means that a fixed prompt and response are used, and at each stage the logit vector is recorded, but instead of sampling a token according to the logit vector, the token in the corresponding sequence location of the fixed sequence is added to the context. In this manner, the probability in the LLM's training distribution of the fixed sequence may be obtained.

Null Prompting: In order to estimate the prior weights w_d of the training distribution, it is useful to repeatedly prompt the LLM under study with a null string (i.e. ""), and extract some small, fixed number of response tokens, providing a small document in response to each null prompt.



Q – SCIENCE
C – AUXILIARY SCIENCES OF HISTORY
G – GEOGRAPHY. ANTHROPOLOGY. RECREATION
H – SOCIAL SCIENCES
N – FINE ARTS
R – MEDICINE
J – POLITICAL SCIENCE
K – LAW
B – PHILOSOPHY. PSYCHOLOGY. RELIGION
T – TECHNOLOGY
F – HISTORY OF THE AMERICAS
D – WORLD HISTORY AND HISTORY OF EUROPE, ASIA, AFRICA, AUSTRALIA, NEW ZEALAND, ETC.
P – LANGUAGE AND LITERATURE
L – EDUCATION
S – AGRICULTURE
Z – BIBLIOGRAPHY. LIBRARY SCIENCE. INFORMATION RESOURCES (GENERAL)
M – MUSIC AND BOOKS ON MUSIC
E – HISTORY OF THE AMERICAS
U – MILITARY SCIENCE
A – GENERAL WORKS
V – NAVAL SCIENCE

Figure 1: Prior weights vs. 1-word prompt (“medicine”) posterior weights.

Dialect Identification: The small documents thus obtained may be classified according to an appropriate classification system, and the relative frequency of occurrence of the classes provide estimates of the w_d. If the documents were produced in response to a repetition of a non-null prompt $\boldsymbol{p}$, then the same technique yields the posterior weights $\hat{w}_d(\boldsymbol{p})$.

3.1 LoC Classification

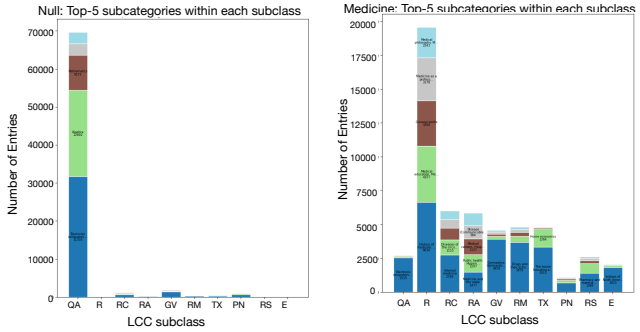
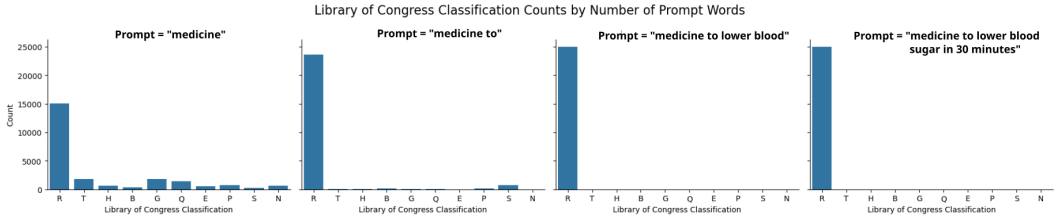


Figure 2: Top 5 categories within top 10 subclasses. Left panel: Null prompt. Right panel: 1-word prompt (“medicine”).

With models trained on natural language, it is not possible to fully test the theory, because we do not know the actual labeled topic dialect distributions, and so (for example) we cannot compute a priori the concentration of posterior weights according to Equation (3). We may however perform a qualitative validation of the theory of prompting and fine-tuning just presented using open-weight pretrained and fine-tuned models. For this purpose, we choose `Meta-Llama-3-8B` (Lla, 2023) and `Meta-Llama-3.1-FoundationAI-SecurityLLM-base-8B` (Lla, 2025), the latter being a fine-tuning of the former using a corpus of cybersecurity-related text.

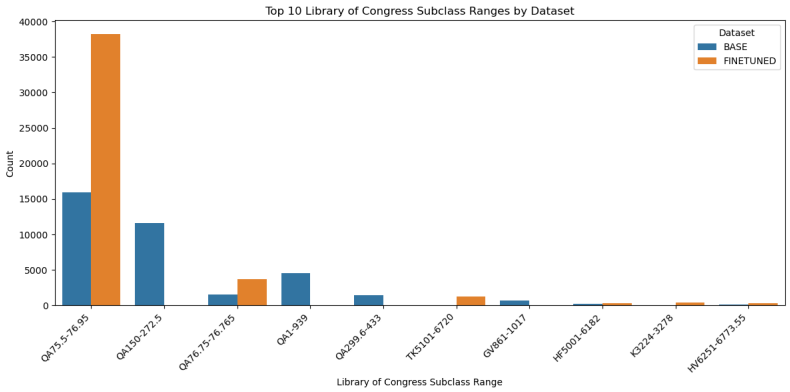
We perform topic classification of text emitted by the models using the Library of Congress Classification (LCC, 2021). This is the hierarchical scheme by which the LoC assigns a unique identifier to each catalogued work. It is technically distinct from the LoC Subject Headings (SHs), which are also topic labels, but which may be multiply-assigned to each work if a work is judged to belong to multiple classes. The LCC is generally assigned to a work based on the SH most relevant to the work. Since we model each document as corresponding to a single topic, the LCC class is the more appropriate document labeling choice.

We estimate model prior dialect weights by null-prompting the models a 100,000 times, eliciting a new 256-token document in each case, and prompting `Claude Opus` 4.8 to choose the most appropriate leaf node classification from the LCC for each document. The estimated prior weight for each class is then simply proportional to the number of documents so classified, normalized so that the weights sum to 1. Posterior weights consequent on prompting are obtained in the same manner.



R – MEDICINE
T – TECHNOLOGY
H – SOCIAL SCIENCES
B – PHILOSOPHY. PSYCHOLOGY. RELIGION
G – GEOGRAPHY. ANTHROPOLOGY. RECREATION
Q – SCIENCE
E – HISTORY OF THE AMERICAS
P – LANGUAGE AND LITERATURE
S – AGRICULTURE
N – FINE ARTS

Figure 3: Progressive concentration of posterior weights with prompt length.



QA75.5-76.95 –Electronic computers. Computer science
QA150-272.5 –Algebra
QA76.75-76.765 –Computer software
QA1-939 –Mathematics
QA299.5-433 –Analysis
TK5101-6720 –Telecommunication
GV861-1017 – Ball games: Baseball, football, golf, etc.
HF5001-6182 –Business
K3224-3278 –Constitutional law: Individual and state
HV6251-6773.55 –Criminology: Crimes and offenses

Figure 4: Effect of fine-tuning pretrained model on cybersecurity text.

3.1.1 Prompting And Posterior Weights

In Figure 1, each pair of bars shows the contrast between the corresponding prior and posterior weights of the pretrained model. The blue, leftmost bar of each pair shows the prior weight resulting from null-prompting. It is clear from these estimated weights that the training data for this model is heavily weighted to science ("Q") topics; the next largest set of topics, medicine ("R") has a factor 10 lower weight.

To observe posterior reweighting, we chose a medicine relevant prompt ("medicine to lower blood sugar in 30 minutes"). The orange, rightmost bar in each pair in the figure displays the topic class weights following the 1-word prompt "medicine." Unsurprisingly, the weight ratio between classes "Q" and "R" is reversed, suggesting that the likelihoods of "medicine" in the dialects of topics in "R" is on average 100 times greater than the likelihoods in dialects of topics in "Q". Interestingly, there are other cases of weight reversals, prominent among which is "E" (History of the Americas). In Figure 2, top subclasses within the top 10 classes are shown, and we can see that most of the posterior weight in class "E" is due to the subclass "Indians of North America" (sic), presumably due to the incidence of the word "medicine" in works on Native American culture.

Figure 3 displays the progressive concentration of posterior weight with increasing prompt length. The concentration is seen to increase very quickly. At 2 words ("medicine to") most classes already have negligible weight, with only class "S" (Agriculture) not completely suppressed, presumably due to the presence training documents on veterinary topics. By 4 words ("medicine to lower blood") it is already extremely unlikely to sample any documents unrelated to medicine, and by the time the full prompt is fed to the model this is overwhelmingly the case. We see here the progressive concentration of topic weights, as off-topic dialects become progressively downweighted. As a consequence, responses are necessarily sampled from increasingly cogent text distributions

3.1.2 FINE-TUNING

In Figure 4, we see the effect on the prior weights of fine-tuning on cybersecurity-related text. Within science topics, we see an unsurprising downweighting of pure mathematics compared to computer science topics. Tthe fine-tuned model also has enhanced emphasis on telecommunications, business, law, and criminology, all topics bearing a clear relationship to the subject of cybersecurity.

3.2 MODEL LANGUAGE

A full validation study of the theory requires that the probability measure $\Pr()$ should be known a priori, rather than learned approximately by the LLM, since this is the only way to verify that the posterior weights shift as predicted by Equation (3). We have therefore designed a small model language that implements a dialect-of-a-common-language distributional structure, to generate "documents" than can be used to train an LLM according to a prescribed distribution.

The full design of the model language is deferred to Appendix 5.2. In summary, it is based on a mixture of D multivariate normal distributions in $\mathbb{R}^N$, each of which is a small perturbation of a common-language multivariate normal distribution. The dialects have covariances and means that are small additive perturbations to the common covariance and mean. The dimension N of the space corresponds to the fixed maximum document length, and each component $x_0, x_1, \ldots, x_{N-1}$ of a vector $\boldsymbol{x}$ sampled from the distribution corresponds to a token in the sequence $(t_0, t_1, \ldots, t_{N-1}) \equiv \boldsymbol{t}$. The tokenization map $\text{Tok} : \mathbb{R} \mapsto V$, $t_\mu = \text{Tok}(x_\mu)$ is simply a partitioning of a large interval $R \in \mathbb{R}$ into $|V|$ equal disjoint bins R_k, $k = 0, \ldots, |V| - 1$, and assigning token IDs according to $x_\mu \in R_k \implies \text{Tok}(x_\mu) = k$. We also define the "untokenizer" function $U : V \mapsto \mathbb{R}$, which assigns to each token ID k a value $U(k)$ that is the midpoint of the interval R_k.

Once the mixture weights $\boldsymbol{w} = [w_0, w_1, \ldots w_{D-1}]$, $\sum_{d=0}^{D-1} w_d = 1$ are chosen, documents $\boldsymbol{t}$ are sampled according to

$$d \sim \text{Categorical}\,(\boldsymbol{w}) \tag{4}$$

$$\boldsymbol{x} \sim \mathcal{N}\,(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d) \tag{5}$$

$$\boldsymbol{t} = \text{Tok} \circ \boldsymbol{x} \tag{6}$$

where $\boldsymbol{\mu}_d$, $\boldsymbol{\Sigma}_d$ are the normal mean and covariance corresponding to dialect d, and where the notation $\text{Tok} \circ \boldsymbol{x}$ means elementwise application of the tokenization map $\text{Tok}()$.

Adopting the notations $\boldsymbol{T}_{<\mu}$, $\boldsymbol{t}_{<\mu}$ to denote the first μ tokens of $\boldsymbol{T}$ and $\boldsymbol{t}$, we use the posterior probability $\Pr(\{\Delta = d\}|\{\boldsymbol{T}_{<\mu} = \boldsymbol{t}_{<\mu}\})$ to classify a text $\boldsymbol{t}_{<\mu}$ according to the dialect that it was sampled from. In §5.2 of the Appendix, we show that

$$\Pr(\{\Delta = d\}|\{\boldsymbol{T}_{<\mu} = \boldsymbol{t}_{<\mu}\}) \approx \frac{w_d \times \pi_{\mathcal{N}}\left(U(\boldsymbol{s}_{<\mu}); \boldsymbol{\mu}_d[:\mu], \boldsymbol{\Sigma}_d[:\mu, :\mu]\right)}{\sum_{d'=0}^{D-1} w_{d'} \times \pi_{\mathcal{N}}\left(U(\boldsymbol{s}_{<\mu}); \boldsymbol{\mu}_{d'}[:\mu], \boldsymbol{\Sigma}_{d'}[:\mu, :\mu]\right)}, \tag{7}$$

where $\pi_{\mathcal{N}}(\cdot; \boldsymbol{\mu}_d[:\mu], \boldsymbol{\Sigma}_d[:\mu, :\mu])$ is the multivariate normal density with mean $\boldsymbol{\mu}_d[:\mu]$ and covariance $\boldsymbol{\Sigma}_d[:\mu, :\mu]$. The indexing notation in, for example, $\boldsymbol{\Sigma}_d[:\mu, :\mu]$ is intended to be suggestive of Python indexing, and means the upper-left $\mu \times \mu$ sector of the matrix $\boldsymbol{\Sigma}_d$. The text $\boldsymbol{t}_{<\mu}$ is assigned the classification of the dialect index d that maximizes this expression.

We choose the hyperparameters $N = 256$, $|V| = 1000$, $D = 30$, $\forall d : w_d = 1/D$ for this study. We choose the $\boldsymbol{\mu}_d$, $\boldsymbol{\Sigma}_d$ randomly, according to criteria that yield 95% accurate classification for $\mu = 256$ using the maximum of $\Pr(\{\Delta = d\}|\{\boldsymbol{T}_{<\mu} = \boldsymbol{t}_{<\mu}\})$ as the classification label. For our numerical experiments, we train a mini-`bert` model (Turc et al., 1908) with 6 attention layers and embedding size 128 on this data. This choice is motivated by the simplicity of the data distribution, which should be assimilable by a small and cheaply-trainable model.

3.2.1 FINE-TUNING AND CATASTROPHIC FORGETTING

Our methodology for distinguishing between the fine-tuning and catastrophic forgetting is described in detail in §5.3 of the Appendix. Briefly, we pre-train a small `bert` model on a corpus of model language data, using batches of $B_{pretrain} = 25$ documents for $E_{pretrain} = 5,000$ iterations. We fine-tune this model on a small corpus sampled from a single dialect $d_0 = 15$ for a varying number of iterations, $E_{finetune}$. At each iteration, a batch of size $B_{finetune} = 25,000$ documents is processed to update the fine-tuned model. We also pretrain a single-dialect model on $B_{single} = 25$ documents

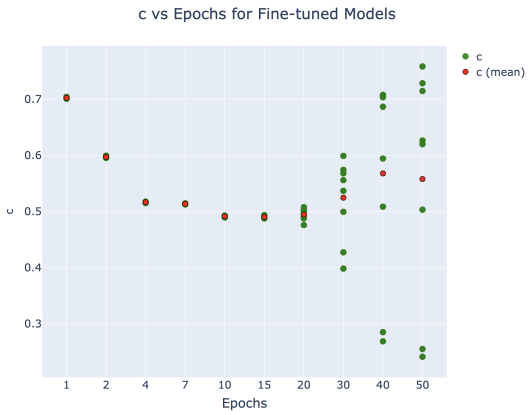


Figure 5: The onset of catastrophic forgetting can be seen at about $E_{finetune} = 20$ in the spreading out of the estimated values of the coefficient c in Equation (8).

per iteration for $E_{single} = 5,000$ iterations. As we discuss in the appendix, in the healthy fine-tuning regime, we expect the probabilities of a document $\boldsymbol{t}$ in the three models to satisfy the relation

$$P_F\left(\{\boldsymbol{T}=\boldsymbol{t}\}\right) = c \times P_P\left(\{\boldsymbol{T}=\boldsymbol{t}\}\right) + (1-c) \times \Pr\left(\{\boldsymbol{T}=\boldsymbol{t}\} \,|\, \{\Delta = d_0\}\right), \tag{8}$$

where $0 < c < 1$. We estimate c by ordinary least squares (OLS) of this relation, using $M = 100$ texts sampled by null-prompting the fine-tuned model, each time coercing the pre-trained and single-dialect model to sample the same text, and computing the probabilities of $\boldsymbol{t}$ in each model by collecting the respective logits. We repeat this estimate $N = 8$ times for each fine-tuning iteration $E_{finetune}$. We expect a small spread in the estimated values of c in the healthy fine-tuning regime. In the catastrophic forgetting regime, we expect that Equation (8) breaks down because the single-dialect distributions learned in pre-training begin to be distorted by fine-tuning for too many epochs, so that no single value of c can describe the relationship between the three models. In this case we expect to see the spread in the estimated values of c grow. The spreading out of c values is thus our signal of the onset of catastrophic forgetting.

The result of the experiment is displayed in Figure 8. The onset of catastrophic forgetting is very clearly distinguishable in the sudden dispersion of c values occurring at $E_{finetune} = 20$. The value attained by c just before this transition is $c \approx 0.5$, indicating that for this model and data, the fine-tuned model can support about 50% weight for the single fine-tuning dialect before significant damage occurs to the other single-dialect distributions, leading to catastrophic forgetting.

4 Discussion

We have exhibited a model-agnostic theory of LLMs in which the LLM itself is represented by a stochastic process indexed by text sequence location and valued in token vocabularies. The role of the LLM in this view is only to learn and sample the probability measure over texts. The burden of explanations is shifted from model structure to data distributional structure, and we have shown that a simple model of natural language as a distribution mixture of topic-specific dialects suffices to provide explanations of LLM prompting and of LLM fine-tuning that are grounded in mathematical probability theory. In fact, all that is required of the LLM is that it should learn a good approximation to the probability measure $\Pr()$. Conditional probability and Bayes' rule are then sufficient to elicit verifiable consequences for prompting and for the fine-tuning/catastrophic forgetting boundary.

It would be interesting to study further whether LLM behavior can in fact be entirely encapsulated by the probability measure $\Pr()$, irrespective of internal computational structure. That structure would then be important to the efficiency and accuracy with which the model learns to approximate $\Pr()$, but could not then be ascribed any higher cognitive processing such as "Artificial General Intelligence" (Bubeck et al., 2023), since such cognition could not presumably be represented by simple sampling of distributions over texts.

Future work in this subject can certainly include the study of the contraction of posterior dialect weights using a model language similar to the one introduced here, so as to actually verify quantitatively that a trained transformer model exhibits posterior dialect re-weighting according to Equation (3).

AI USE STATEMENT

In this work, generative AI models were used for a single purpose: to generate the LCC class assignments of texts elicited by prompting an LLM, as described in §3.1. No generative AI was utilized for the purpose of helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implement methods, assisting with translation, cleaning or reformatting a dataset, supporting qualitative or thematic data analysis, or interpreting results. No part of the text of this manuscript was generated using AI, although a late AI-generated review of the manuscript suggested a useful missing reference to related work.

ACKNOWLEDGMENTS

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

REFERENCES

Library of congress classification outline, 2021. URL https://www.loc.gov/catdir/cpso/lcco/.

Meta-llama-3-8b model card, 2023. URL https://huggingface.co/meta-llama/Meta-Llama-3-8B.

Meta-llama-3.1-foundationai-securityllm-base-8b model card, 2025. URL https://huggingface.co/fdtn-ai/Foundation-Sec-8B.

Patrick Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995. ISBN 9780471007104.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Tinglong Dai and Terry Taylor. Designing enterprise ai systems: Hallucination, creativity, and moral hazard. *Creativity, and Moral Hazard (December 31, 2025)*, 2025.

Marwa Khairallah. Governance challenges and ethical risks of agentic ai in education: Accountability, hallucination, bias amplification, and control mechanisms. In *2026 International Symposium of Systems, Advanced Technologies and Knowledge (ISSATK)*, pp. 1–8. IEEE, 2026.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Carlos Olea, Holly Tucker, Jessica Phelan, Cameron Pattison, Shen Zhang, Maxwell Lieb, Doug Schmidt, and Jules White. Evaluating persona prompting for question answering tasks. In *Proceedings of th e 10th international conference on artificial intelligence and soft computing, Sydney, Australia*, 2024.

Dan Shiebler, Bruno Gavranović, and Paul Wilson. Category theory in machine learning. *arXiv preprint arXiv:2106.07032*, 2021.

Shilpika Shilpika, Carlo Graziani, Bethany Lusch, Venkatram Vishwanath, and Michael E Papka. Probabilistic attribution for large language models. *arXiv preprint arXiv:2605.21726*, 2026.

Yunlin Su and Song Y. Yan. *Principles of Compilers: A New Approach to Compilers Including the Algebraic Method*. Springer Berlin, Heidelberg, 2011. ISBN 978-3-642-20834-8.

Janet Todd. *The Cambridge Introduction to Jane Austen*. Cambridge introductions to literature. Cambridge University Press, 2006. ISBN 9780521858069.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models (2019). *arXiv preprint arXiv:1908.08962*, 1908.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ike Vayansky and Sathish A.P. Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, 2020. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2020.101582. URL https://www.sciencedirect.com/science/article/pii/S0306437920300703.

Yifei Wang. A comparative analysis of model agnostic techniques for explainable artificial intelligence. *Research Reports on Computer Science*, pp. 25–33, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. In *International Conference on Learning Representations*, volume 2024, pp. 44694–44713, 2024.

5 Appendix

5.1 Stochastic Process Theory of LLMs

An LLM may be naturally viewed as *a stochastic process, indexed by natural numbers (sequence position), and valued in finite sets (token vocabularies)*. This allows us to describe the distribution of language sequences in its native space, instead of in a model embedding Euclidean space of arbitrary dimension. We introduce the theory of such processes, following (Billingsley (1995) pp 27–30).

A stochastic process is a collection of random variables T_μ indexed by some set J (i.e. $\mu \in J$) over a probability space $(\Omega, \mathcal{F}, \Pr)$. Ω is the sample space, $\mathcal{F}$ is the σ-algebra whose elements are measurable subsets of Ω, and $\Pr$ is the probability measure over the space.

In the case of LLMs:

$J = \mathcal{N}_0$, the natural numbers including 0.

Define $\Omega_S = \{s : s = (t_0, t_1, t_2, \ldots), t_\mu \in V\}$, the set of all infinite sequences of tokens $t_\mu \in V$, $\forall \mu \in \mathcal{N}_0$, where V is the token vocabulary, a discrete finite set of cardinality $|V|$.

Also define $\Omega_D = \{d : d \in \mathcal{N}_0, d < D\}$, the set of topic dialect indices.

The sample space is $\Omega = \Omega_S \times \Omega_D$. A sample point $\omega \in \Omega$ is a pair $\omega = (s, d)$, $s \in \Omega_s$, $d \in \Omega_D$ that associates a single topic with a sample sequence.

$\Delta : \Omega \mapsto \Omega_D$, $\Delta(s, d) = d$ is the projection onto dialect index. When the probability measure is defined, it becomes a random variable, which is to say a measurable function.

$T_\mu : \Omega \mapsto V$, $T_\mu(s, d) = t_\mu$ is the projection of the sequence $s = (t_0, t_1, \ldots)$ onto the token t_μ. When the probability measure is defined, it becomes an indexed collection of random variables, which is to say a collection of measurable functions.

Define a *finite* index set $I = \{i_0, i_1, \ldots, i_{|I|-1}\} \subset \mathcal{N}_0$, $|I| < \infty$, and the finite subsequence $\boldsymbol{t}_I \equiv (t_{i_0}, t_{i_1}, \ldots, t_{i_{|I|-1}})$, as well as the random variable $\boldsymbol{T}_I \equiv (T_{i_0}, \ldots, T_{i_{|I|-1}})$.

The collection of $\mathcal{F}_0$ of all sets $m(\boldsymbol{t}_I) = \{\omega \in \Omega : \boldsymbol{T}_I(\omega) = \boldsymbol{t}_I\}$, $\forall I, \boldsymbol{t}_I$, as well as their complements and finite unions, is a field.

The σ-field $\mathcal{F} = \sigma(\mathcal{F}_0)$ generated by $\mathcal{F}_0$ is the *event space*: it is the collection of *measurable sets*.

The $\boldsymbol{T}_I(\omega)$ are measurable functions: their inverse images $\{\boldsymbol{T}_I = \boldsymbol{t}_I\} \equiv \{\omega \in \Omega : \boldsymbol{T}_I(\omega) = \boldsymbol{t}_I\}$ are in $\mathcal{F}$. Similarly, $\Delta(\omega)$ is measurable, as its inverse images $\{\Delta = d\} \equiv \{\omega \in \Omega : \Delta(\omega) = d\}$ are also in $\mathcal{F}$.

5.1.1 THE LLM PROBABILITY MEASURE

The measure $Pr : \mathcal{F} \mapsto [0, 1]$ is the *probability measure*, satisfying $\Pr(\emptyset) = 0$, $\Pr(\Omega) = 1$, and $\Pr(\cup_\alpha A_\alpha) = \sum_\alpha \Pr(A_\alpha)$ for disjoint $A_\alpha \in \mathcal{F}$. Interestingly, this measure is not specified in closed form, but rather *it is learned by the LLM from its training data corpus*. The LLM samples that distribution on a token-by-token basis to produce responses. The log-probabilities for the next token conditioned on previous tokens may be read off from the model as they are sampled. From these "logits" one may in fact reconstruct the probability of arbitrary sub-sequences $\{\boldsymbol{T}_J = \boldsymbol{t}_J\}$ conditioned on other arbitrary sub-sequences $\{\boldsymbol{T}_K = \boldsymbol{t}_K\}$, $J \cap K = \emptyset$. Which is to say, *the entire approximation to* $\Pr()$ *learned by the LLM in training is accessible to us*. This fact was used in (Shilpika et al., 2026) to construct sensitivity measures of LLM responses to individual prompt tokens, but not proved in the most general case. Here we supply that proof.

Consider a fixed token sequence $\boldsymbol{t} = (t_0, \ldots, t_{M-1})$ of length M. Define the index set $I_{<\mu} \equiv \{0, 1, 2, \ldots \mu - 1\}$, so that, for example $\boldsymbol{t} = \boldsymbol{t}_{I_{<M}}$.

We may recover the LLM next token probabilities $\Pr\left(\{T_\mu = t_\mu\} | \{\boldsymbol{T}_{I_{<\mu}} = \boldsymbol{t}_{I_{<\mu}}\}\right)$ from a softmax of the logits obtained by "coercive" sampling, which is to say that at stage μ one adds t_μ to the LLM's context instead of a token s_μ sampled from the categorical distribution $\Pr\left(\{T_\mu = s_\mu\} | \{\boldsymbol{T}_{I_{<\mu}} = \boldsymbol{t}_{I_{<\mu}}\}\right)$, as is usual. From the softmaxed logits, we may compute

$$\Pr\left(\{\boldsymbol{T}_{I_{<M}} = \boldsymbol{t}\}\right) = \prod_{\mu=0}^{M-1} \Pr\left(\{T_\mu = t_\mu\} | \{\boldsymbol{T}_{I_{<\mu}} = \boldsymbol{t}_{I_{<\mu}}\}\right) \tag{9}$$

for an arbitrary sequence $\boldsymbol{t}$.

Now partition the index set $I_{<M}$ into three arbitrary disjoint index sets J, K, L, so that $J \cap K = K \cap L = L \cap J = \emptyset$, $J \cup K \cup L = I_{<M}$. We may compute the general conditional probability $\Pr\left(\{\boldsymbol{T}_J = \boldsymbol{t}_J\} \,|\, \{\boldsymbol{T}_K = \boldsymbol{t}_K\}\right)$:

$$\begin{aligned}\Pr\left(\{\boldsymbol{T}_J = \boldsymbol{t}_J\} \,|\, \{\boldsymbol{T}_K = \boldsymbol{t}_K\}\right) &= \frac{\Pr\left(\{\boldsymbol{T}_J = \boldsymbol{t}_J\} \cap \{\boldsymbol{T}_K = \boldsymbol{t}_K\}\right)}{\Pr\left(\{\boldsymbol{T}_K = \boldsymbol{t}_K\}\right)} \\ &= \frac{\sum_{\boldsymbol{t}_L \in V^{|L|}} \Pr\left(\{\boldsymbol{T}_{I_{<M}} = \boldsymbol{t}\}\right)}{\sum_{\boldsymbol{t}_L \in V^{|L|}} \sum_{\boldsymbol{t}_J \in V^{|J|}} \Pr\left(\{\boldsymbol{T}_{I_{<M}} = \boldsymbol{t}\}\right)}.\end{aligned} \tag{10}$$

Since the summand is computable from Equation (9), the general conditional is also computable. The summations in Equation (10) may be intractably large if $|V|$ and either $|J|$ or $|L|$ are large, but they are computable in principle, which establishes the assertion that the full probability measure $\Pr()$ is recoverable from the LLM's next-token logits.

5.2 MODEL LANGUAGE DESIGN

The mixture-of-dialects model is based on a normal mixture in $\mathbb{R}^N$, with one mixture component per dialect. The number of distributions in the normal mixture (i.e. the number of dialects) is a hyperparameter, D. The dialect weights are denoted w_d, $d = 0, \ldots, D-1$. Vectors sampled from the distribution are converted to N-token documents, where N is a hyperparameter.

Tokenization is performed by partitioning a large interval of the real number line $R \subset \mathbb{R}$ into $|V|$ equal-length disjoint intervals $R_t \subset R$, $t = 0, \ldots, |V| - 1$, $R_t \cap R_{t'} = \emptyset$ if $t \neq t'$, $\cup_t R_t = R$, to represent the token vocabulary V. Each sampled vector yields one token per vector component, corresponding to the bin in which the component's value falls. The map $\text{Tok} : \mathbb{R} \mapsto V$ represents the tokenization, i.e. $x \in R_t \implies \text{Tok}(x) = t$. Vector components falling to the left or right of R are mapped to two "unknown" tokens. The document tokenizer is a function $T : \mathbb{R}^N \mapsto V^N$, $T(\boldsymbol{x}) = \text{Tok} \odot \boldsymbol{x}$, where "$\odot$" means elementwise application of the function.

It is convenient to define an "untokenizer" function $U : V^N \mapsto \mathbb{R}^N$, which maps each token to the corresponding vector component, whose assigned value is the center of the corresponding tokenizer bin, so that $T\left(U\left(\boldsymbol{s}\right)\right) = \boldsymbol{s}$.

The dialect distributions are constructed as perturbations of a common language distribution. We specify the common language distribution as the tokenization of a zero-mean normal distribution with a random covariance $\boldsymbol{\Sigma}_{Common} = \boldsymbol{A}^\top \boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is a random matrix whose elements are sampled from the standard normal distribution, $\boldsymbol{A} \sim \mathcal{N}(\boldsymbol{0}_{N \times N}, \boldsymbol{I}_{(N \times N) \times (N \times N)})$. The distribution corresponding to the common language is thus the tokenization of $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{Common})$.

Each of the D dialects is specified by specifying a "small" mean vector $\boldsymbol{\mu}_d$ and a "small" additive perturbation $\boldsymbol{C}_d$ to $\boldsymbol{\Sigma}_{Common}$, where the smallness is chosen in a manner to be described below in order to guarantee that the dialect distributions are all small yet identifiable perturbations of the common language. The dialect d is associated with a document-valued random variable $\boldsymbol{S}_d \in V^N$, $\boldsymbol{S}_d = T(\boldsymbol{X}_d)$, $\boldsymbol{X}_d \sim \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$, where $\boldsymbol{\Sigma}_d \equiv \boldsymbol{\Sigma}_{Common} + \boldsymbol{C}_d$.

A document is sampled according to Equations (4–6), that is, by first sampling a dialect d from the categorical distribution with probabilities w_d, then sampling a vector $\boldsymbol{x}$ from the multivariate normal distribution corresponding to that dialect, and tokenizing the result.

5.2.1 Document Probability

We may calculate the probability of a document $\boldsymbol{s}$ token-by-token. It is given by

$$\Pr\left(\{\boldsymbol{T}_{<N} = \boldsymbol{s}\}\right) = \sum_{d=0}^{D-1} w_d \Pr\left(\{\boldsymbol{T}_{<N} = \boldsymbol{s}\} \,|\, \{\Delta = d\}\right), \tag{11}$$

$$\Pr\left(\{\boldsymbol{T}_{<N} = \boldsymbol{s}\} \,|\, \{\Delta = d\}\right) = \prod_{\mu=0}^{N-1} \Pr\left(\{T_\mu = s_\mu\} \,|\, \{\boldsymbol{T}_{<\mu} = \boldsymbol{s}_{<\mu}\} \cap \{\Delta = d\}\right)$$

$$\Pr\left(\{T_\mu = s_\mu\} \,|\, \{\boldsymbol{T}_{<\mu} = \boldsymbol{s}_{<\mu}\} \cap \{\Delta = d\}\right) \approx \int_{R_{s_\mu}} dx_\mu \, \pi\left(x_\mu | U(\boldsymbol{s}_{<\mu}), d\right). \tag{12}$$

Here, s_μ is the μ-th token of the sequence $\boldsymbol{s}$, while $\boldsymbol{s}_{<\mu}$ is the length-μ sequence of tokens preceding μ. $\pi\left(x_\mu | U(\boldsymbol{s}_{<\mu}), d\right)$ is the normal theory conditional density for x_μ given the vector of bin centers $U(\boldsymbol{s}_{<\mu})$:

$$\pi\left(x_\mu | U(\boldsymbol{s}_{<\mu}), d\right) = \pi_{\mathcal{N}}\left(x_\mu; m_{d,\mu}, \sigma^2_{d,\mu}\right) \tag{13}$$

$$m_{d,\mu} \equiv \boldsymbol{\mu}_d[\mu] + \boldsymbol{\Sigma}_d[\mu, :\mu]\boldsymbol{\Sigma}_d[:\mu, :\mu]^{-1}\left(U(\boldsymbol{s}_{<\mu}) - \boldsymbol{\mu}[:\mu]\right) \tag{14}$$

$$\sigma^2_{d,\mu} \equiv \boldsymbol{\Sigma}_d[\mu, \mu] - \boldsymbol{\Sigma}_d[\mu, :\mu]\boldsymbol{\Sigma}_d[:\mu, :\mu]^{-1}\boldsymbol{\Sigma}_d[:\mu, \mu]. \tag{15}$$

The integrals over the bins R_{s_μ} may be carried out in closed form, using the error function. There is an approximation here, in that we condition on the vector $U(\boldsymbol{s}_{<\mu}) \in \mathbb{R}^\mu$, rather than on $T^{-1}(\boldsymbol{s}_{<\mu})$ (i.e. the region in $\mathbb{R}^N$ of vectors whose first μ elements are mapped by $T(\cdot)$ to $\boldsymbol{s}_{<\mu}$). If the width of the tokenization intervals R_s is small compared to the typical variances in the distribution, this is a negligible error.

In fact, one may take the approximation further for the sake of computational efficiency. We may write

$$\Pr\left(\{\boldsymbol{T}_{<N} = \boldsymbol{s}\} \,|\, \{\Delta = d\}\right) \approx \pi_{\mathcal{N}}\left(U(\boldsymbol{s}); \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d\right) \times \left(|R|/|V|\right)^N, \tag{16}$$

where $|R|$ is the length of the interval R, and $\pi_{\mathcal{N}}\left(\cdot; \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d\right)$ is the normal density corresponding to dialect d. This equation is approximately correct if the tokenization interval length $|R|/|V|$ is much smaller than the square root of any eigenvalue of $\boldsymbol{\Sigma}_d$. It states that the probability of a sampled document $\boldsymbol{s}$ in the dialect d is roughly equal to the normal density $\pi_{\mathcal{N}}\left(\cdot; \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d\right)$ evaluated at the untokenized location $U(\boldsymbol{s}) \in \mathbb{R}^N$, multiplied by the volume of the little N-dimensional box containing all the points in $\mathbb{R}^N$ that tokenize to $\boldsymbol{s}$. In other words,

$$\Pr\left(\{\boldsymbol{T}_{<N} = \boldsymbol{s}\} \,|\, \{\Delta = d\}\right) \approx \kappa \left(\det \boldsymbol{\Sigma}_d\right)^{-1/2} \exp\left[-\frac{1}{2}\left(U(\boldsymbol{s}) - \boldsymbol{\mu}_d\right)^\top \boldsymbol{\Sigma}_d^{-1}\left(U(\boldsymbol{s}) - \boldsymbol{\mu}_d\right)\right], \tag{17}$$

where κ is a normalization constant that is independent of d.

We may apply the same approximation to compute the probability of the first μ tokens in a document (e.g. of a prompt):

$$\begin{aligned}\Pr\left(\{\boldsymbol{T}_{<\mu} = \boldsymbol{s}_{<\mu}\}\right) &= \sum_{d=0}^{D-1} w_d \Pr\left(\{\boldsymbol{T}_{<\mu} = \boldsymbol{s}_{<\mu}\} \,|\, \{\Delta = d\}\right) \\ &\approx \left(|R|/|V|\right)^\mu \sum_{d=0}^{D-1} w_d \pi_{\mathcal{N}}\left(U(\boldsymbol{s}_{<\mu}); \boldsymbol{\mu}_d[:\mu], \boldsymbol{\Sigma}_d[:\mu, :\mu]\right).\end{aligned} \tag{18}$$

We may also compute the conditional probability of a length $N-\mu$ response $\boldsymbol{r}$, conditioned on a length μ prompt:

$$\begin{aligned}\Pr\left(\{\boldsymbol{R}=\boldsymbol{r}\}\,|\,\{\boldsymbol{P}=\boldsymbol{p}\}\right) &= \sum_{d=0}^{D-1} w_d \Pr\left(\{\boldsymbol{R}=\boldsymbol{r}\}\,|\,\{\boldsymbol{P}=\boldsymbol{p}\}\cap\{\Delta=d\}\right) \\ &\approx \left(|R|/|V|\right)^{N-\mu} \sum_{d=0}^{D-1} w_d \pi_{\mathcal{N}}\left(U(\boldsymbol{r}); \boldsymbol{m}_{d,\mu:}, \boldsymbol{C}_{d,\mu:}\right), \end{aligned} \tag{19}$$

where

$$\boldsymbol{m}_{d,\mu:} \equiv \boldsymbol{\mu}_d[\mu:] + \boldsymbol{\Sigma}_d[\mu:,:\mu]\boldsymbol{\Sigma}_d[:\mu,:\mu]^{-1}\left(U(\boldsymbol{p}) - \boldsymbol{\mu}[:\mu]\right) \tag{20}$$

$$\boldsymbol{C}_{d,\mu:} \equiv \boldsymbol{\Sigma}_d[\mu:,\mu:] - \boldsymbol{\Sigma}_d[\mu:,:\mu]\boldsymbol{\Sigma}_d[:\mu,:\mu]^{-1}\boldsymbol{\Sigma}_d[:\mu,\mu:]. \tag{21}$$

5.2.2 Dialect Identification

We may use Equation(17) to *classify* documents $\boldsymbol{s}$ by the dialect from which they are sampled. Suppose that we have a corpus of C documents $\boldsymbol{s}_a$, $a=0,\ldots,C-1$, and we know that samples in the corpus from dialect d have weight w_d (i.e. the fraction of such documents in the corpus is w_d). Then the probability that $\boldsymbol{s}_a$ was sampled from dialect d is given by

$$\begin{aligned}\Pr\left(\{\Delta=d\}\,|\,\{\boldsymbol{T}_{<N}=\boldsymbol{s}_a\}\right) &= \alpha \Pr\left(\{\boldsymbol{T}_{<N}=\boldsymbol{s}_a\}\,|\,\{\Delta=d\}\right) \times \Pr(\{\Delta=d\}) \\ &= \alpha w_d \times \Pr\left(\{\boldsymbol{T}_{<N}=\boldsymbol{s}_a\}\,|\,\{\Delta=d\}\right), \end{aligned} \tag{22}$$

where α is another normalization constant. Combining Equations (17) and (22), we find

$$\Pr\left(\{\Delta=d\}\,|\,\{\boldsymbol{T}_{<N}=\boldsymbol{s}_a\}\right) = \frac{w_d \times \pi_{\mathcal{N}}\left(U(\boldsymbol{s}); \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d\right)}{\sum_{d'=0}^{D-1} w_{d'} \times \pi_{\mathcal{N}}\left(U(\boldsymbol{s}); \boldsymbol{\mu}_{d'}, \boldsymbol{\Sigma}_{d'}\right)}. \tag{23}$$

This formula can serve to classify documents by dialect, by choosing the dialect with the highest probability.

Similarly, if we have the initial μ tokens of a document (e.g. a prompt) $s_{<\mu}$, we have

$$\Pr\left(\{\Delta=d\}\,|\,\{\boldsymbol{T}_{<\mu}=\boldsymbol{s}_{<\mu}\}\right) = \frac{w_d \times \pi_{\mathcal{N}}\left(U(\boldsymbol{s}_{<\mu}; \boldsymbol{\mu}_d[:\mu], \boldsymbol{\Sigma}_d[:\mu,:\mu]\right)}{\sum_{d'=0}^{D-1} w_{d'} \times \pi_{\mathcal{N}}\left(U(\boldsymbol{s}_{<\mu}); \boldsymbol{\mu}_{d'}[:\mu], \boldsymbol{\Sigma}_{d'}[:\mu,:\mu]\right)}. \tag{24}$$

5.2.3 Dialect Constants

We may now finally say how we may specify the dialect design constants $\boldsymbol{\mu}_d$ and the $\boldsymbol{\Sigma}_d$. The criterion is that each dialect should be identifiable. This means that if a document $\boldsymbol{s}$ is sampled from dialect d, then we should require, on average, that $\Pr\left(\{\Delta=d\}\,|\,\{\boldsymbol{T}_{<\mu}=\boldsymbol{s}_{<\mu}\}\right) > 1-\epsilon$, for some choice of ϵ and μ. We might choose $\mu=16$ and $\epsilon=0.1$, or $\epsilon=0.05$ to obtain 90% or 95% identifiability with 16 initial tokens.

Given some choice of dialect parameters, we sample M documents $\boldsymbol{s}$ from each dialect ($M \gtrsim 1000$, say), set all the w_d equal, and compute for each one its estimated dialect d from the max of $\Pr\left(\{\Delta=d\}\,|\,\{\boldsymbol{T}_{<N}=\boldsymbol{s}\}\right)$. We adjust the dialect parameters until this identification has minimum accuracy $1-\epsilon$, meaning that no dialect emits samples that get misclassified at a rate greater than ϵ.

To obtain the $\boldsymbol{\mu}_d$, we first sample a set of $\boldsymbol{m}_d$ from the common language,

$$\boldsymbol{m}_d \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_{Common}\right)$$

and set $\boldsymbol{\mu}_d = \beta \boldsymbol{m}_d$. The adjustable parameter $0 < \beta \lesssim 1$ ensures that no dialect receives a mean that shifts it too far from the common language.

We also sample matrices

$$\boldsymbol{A}_d \sim \mathcal{N}(\boldsymbol{0}_{N\times N}, \boldsymbol{I}_{(N\times N)\times(N\times N)})$$

and set

$$\boldsymbol{\Sigma}_d = \boldsymbol{\Sigma}_{Common} + \beta \boldsymbol{A}_d^\top \boldsymbol{A}_d.$$

This makes all the dialect parameters a function of the single parameter β. As $\beta \to 0$, all dialects become identical to the common language, and they all diverge from one another as β grows in magnitude. We adjust the value of β until we obtain the desired degree of identifiability.

5.3 Detecting The Fine-Tuning And Catastrophic Forgetting Regimes

We posited in §2.2 that the healthy fine-tuning regime obtains when fine-tuning on documents from a single topic dialect produces a modification of the pre-trained model's dialect weights, with only minor perturbations to its learned single-dialect distributions, while the catastrophic forgetting regime is reached when the learned single-dialect distributions are compromised by fine-tuning for too many training epochs.

To test this idea, we introduce a simplified model of fine-tuning in the "healthy" regime. In the healthy fine-tuning regime, the re-weighting of the dialects and preservation of the single-dialect distributions can be approximately modeled as pre-training with a corpus in which the specialized task has an increased number of documents compared to the pre-training corpus. In this simple model, we assume that the model's weights of dialects not involved in fine-tuning behave as if the number of pre-training documents sampled from those dialects is unchanged.

Suppose the specialized dialect is indexed $d = d_0$. The pretraining corpus has n_d documents corresponding to dialect d. The simplified model represents the fine-tuned model as a pre-trained model with an additional δn documents from dialect d_0. It follows that the pre-trained dialect weights are $\alpha_d^P = n_d/n$, where $n \equiv \sum_d n_d$. The fine-tuned dialect weights are $\alpha_d^F = (n_d + \delta n \times \delta_{dd_0})/(n + \delta n)$.

Let the probability of a text $\boldsymbol{t}$ be $P_P(\{\boldsymbol{T} = \boldsymbol{t}\})$ in the pre-trained model, and $P_F(\{\boldsymbol{T} = \boldsymbol{t}\})$ in the fine-tuned model. Using the simplified model, we may write

$$\begin{aligned} P_F\left(\{\boldsymbol{T} = \boldsymbol{t}\}\right) &= \sum_{d=0}^{D-1} \alpha_d^F \Pr\left(\{\boldsymbol{T} = \boldsymbol{t}\} \,|\, \{\Delta = d\}\right) \\ &= c \times P_P\left(\{\boldsymbol{T} = \boldsymbol{t}\}\right) + (1-c) \times \Pr\left(\{\boldsymbol{T} = \boldsymbol{t}\} \,|\, \{\Delta = d_0\}\right), \end{aligned} \tag{25}$$

where $0 < c \equiv \frac{n}{n+\delta n} < 1$.

In general, we do not know the value of c consequent on some number of fine-tuning epochs. We may however estimate c by ordinary least-squares regression. According to Equation (25), the probability of $\boldsymbol{t}$ according to the fine-tuned model is a convex combination of the probability of $\boldsymbol{t}$ according to the pre-trained model and the probability of $\boldsymbol{t}$ according to a single-dialect model trained only on texts sampled from dialect d_0. Since we have an inexpensive-to-train `bert` model and an inexhaustible supply of data from the model language, we may sample a pre-training corpus just from dialect d_0 and train a third model just on that data.

The OLS strategy to make a single determination of c is to null-prompt the fine-tuned model M times, each time recording the resulting document $\boldsymbol{t}_m$, $m = 0, \ldots, M-1$ and recording the logits to compute $P_P\left(\{\boldsymbol{T} = \boldsymbol{t}_m\}\right)$. Next, coercively sample the pre-trained model to compute $P_P\left(\{\boldsymbol{T} = \boldsymbol{t}_m\}\right)$, and similarly coercively sample the single-dialect pre-trained model to obtain $\Pr\left(\{\boldsymbol{T} = \boldsymbol{t}_m\} \,|\, \{\Delta = d_0\}\right)$. The OLS objective is then

$$S(c) \equiv \sum_{m=0}^{M-1} \left[c \times P_P\left(\{\boldsymbol{T} = \boldsymbol{t}_m\}\right) + (1-c) \times \Pr\left(\{\boldsymbol{T} = \boldsymbol{t}_m\} \,|\, \{\Delta = d_0\}\right) - P_F\left(\{\boldsymbol{T} = \boldsymbol{t}_m\}\right)\right]^2. \tag{26}$$

Minimizing this with respect to c, we straightforwardly obtain the usual OLS result. Defininining for brevity $x_m \equiv P_P\left(\{\boldsymbol{T} = \boldsymbol{t}_m\}\right)$, $y_m \equiv \Pr\left(\{\boldsymbol{T} = \boldsymbol{t}_m\} \,|\, \{\Delta = d_0\}\right)$, $z_m \equiv P_F\left(\{\boldsymbol{T} = \boldsymbol{t}_m\}\right)$, we have

$$c = \frac{\sum_{m=0}^{M-1}(x_m - y_m)(z_m - y_m)}{\sum_{m=0}^{M-1}(x_m - y_m)^2}. \tag{27}$$

We repeat this process N times, obtaining N determinations of c. If Equation (25) is approximately correct, these values should exhibit relatively small scatter about some mean c. On the other hand, in the catastrophic forgetting regime, the fine-tuning damages the single-dialect distributions learned in pre-training, and Equation (25) is no longer valid. In this case, we should observe an increase in the spread of c values, because no single c can capture the relationship between the three models. Our measure of the onset of catastrophic forgetting is therefore constructed from the spread in the N estimates of c.