Extracting memorized pieces of (copyrighted) books from open-weight language models

A. Feder Cooper [1] **Aaron Gokaslan** [2] **Ahmed Ahmed** [1] **Amy B. Cyphert** [3] **Christopher De Sa** [4]
Mark A. Lemley [5] **Daniel E. Ho** [5] **Percy Liang** [1]

Abstract

Plaintiffs and defendants in copyright lawsuits over generative AI often make sweeping, opposing claims about the extent to which large language models (LLMs) have memorized plaintiffs' protected expression. Drawing on adversarial ML and copyright law, we show that these polarized positions dramatically oversimplify the relationship between memorization and copyright. To do so, we leverage a recent probabilistic extraction technique to extract pieces of the `Books3` dataset from 17 open-weight LLMs. Through numerous experiments, we show that it's possible to extract substantial parts of at least some books from different LLMs. This is evidence that these LLMs have memorized the extracted text; this memorized content is copied inside the model parameters. But the results are complicated: the extent of memorization varies both by model and by book. With our specific experiments, we find that the largest LLMs don't memorize most books—either in whole or in part. However, we also find that LLAMA 3.1 70B memorizes some books, like *Harry Potter and the Sorcerer's Stone* and *1984*, almost entirely. In fact, *Harry Potter* is so memorized that, using a seed prompt consisting of just the first line of chapter 1, we can deterministically generate the entire book near-verbatim. We discuss why our results have significant implications for copyright cases, though not ones that unambiguously favor either side.

1. Introduction

In the dozens of pending copyright suits over training LLMs, the opposing parties tend to present their interpretation of the technical operation of models in simplified terms. Plaintiffs say LLMs are just giant (infringing) copy machines that store their works and recombine them in their outputs (Kadrey v. Meta Platforms, Inc.). Defendants say LLMs merely contain linguistic relationships—"statistical correlations" (Concord Music Group, Inc. v. Anthropic PBC)—and don't copy the plaintiffs' works. The situation is more complicated than either side suggests.

Appreciating why requires a deeper understanding of training-data extraction, training-data memorization, and the relationship between the two (Section 2). While extraction refers to reconstructing specific training data from a model's generated outputs, memorization is broader: it involves reconstructing specific training data by examining the model "through any means" (Cooper et al., 2023, Glossary). It's an uncontroversial statement in machine learning (ML) to say that the extraction of a piece of training data implies that the model has memorized that piece of training data (Carlini et al., 2023b; 2021; Schwarzschild et al., 2024)—that extraction is evidence of memorization *inside* the model. As Carlini explains, when a sufficiently large and unique piece of training data is extracted, "the only possible explanation is that the model has somewhere internally stored [that piece of training data]. There just is no other explanation; it can't be [generated] due to chance" (Carlini, 2025). As Cooper and Grimmelmann note, "in order to be able to extract memorized content from a model at generation time, that memorized content must be encoded in the model's parameters. There is nowhere else it could be. A model is not a magical portal that pulls fresh information from some parallel universe into our own" (Cooper & Grimmelmann, 2024, p. 25).

Memorization may have significant consequences for ongoing copyright litigation. Extraction generates a "copy" of training data, but it also demonstrates the existence of a "copy" of that training data is memorized inside the model itself. The model being a "copy" (in a technical sense that copyright cares about) has important implications (Sections 3 & 6). Notably, the models themselves could be deemed infringing copies of the training data they've memorized (Cooper & Grimmelmann, 2024; Lee et al., 2023b). Copyright law offers the destruction of infringing

[1]Stanford University [2]Cornell Tech [3]West Virginia University [4]Cornell University [5]Stanford Law School. Correspondence to: A. Feder Cooper <afedercooper@gmail.com>, Mark A. Lemley <mlemley@law.stanford.edu>.

Published at ICML 2025 Workshop on Reliable and Responsible Foundation Models.

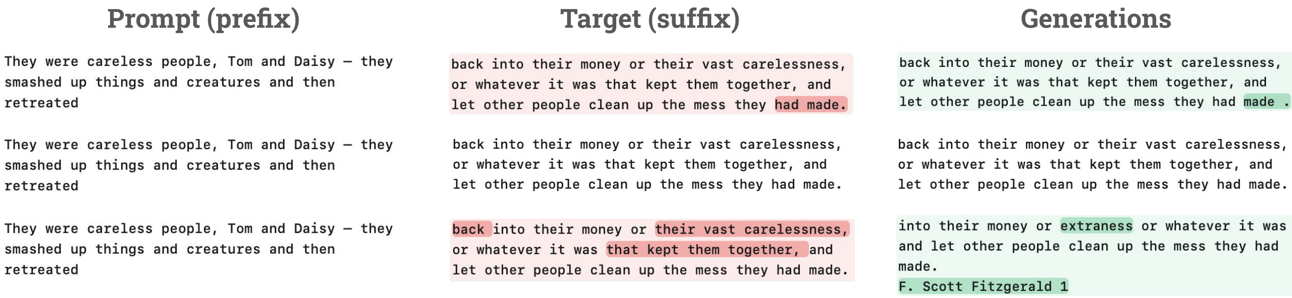


Figure 1: Generating the exact completion of a quote from *The Great Gatsby* (Fitzgerald, 1925) with Llama 1 30B.

materials as a remedy. So, just as courts have ordered the destruction of bootleg DVDs, a court could order the destruction of infringing models (Lee et al., 2023b; Samuelson, 2023; Wilf-Townsend, 2024).

When seeking such remedies in their lawsuit complaints, plaintiffs often cite ML research papers that demonstrate training-data extraction from open-weight models and production systems. But this reasoning is flawed for three key reasons. First, citing a research paper that extracts, for example, an alphabetized list of U.S. states from ChatGPT (Nasr et al., 2023) is *not* evidence that ChatGPT has memorized a plaintiff's own work (Carlini, 2025; Cooper & Grimmelmann, 2024). There is no evidence that most training data is memorized—especially not in high-quality, contemporary LLMs. Second, while most ML research reports average extraction rates for a given extraction attack (i.e., for a particular population of prompts and threat model), averages aren't necessarily relevant to specific copyright infringement claims (Section 3). Instead, as others have also noted (Lee et al., 2023b; Cooper & Grimmelmann, 2024), it may be more relevant to show the extent to which the *specific* model in question has memorized the plaintiff's *specific* copyrighted work. Third, extraction methods may require thousands of runs of the same prompt to generate a given 50-token sequence extracted from a given book. That number is smaller than chance would suggest, so it's evidence for memorization (Section 2); but extraction like this is *not* evidence that any user is actually likely in practice to generate even 50 tokens from that book, much less most or all of that book.

We study the extent to which open-weight models have memorized specific, verbatim pieces of text in the `Books3` dataset: the (now notorious) torrented corpus of nearly 200,000 books. `Books3` is in The Pile (Gao et al., 2020)—an LLM pre-training dataset that has been the focus of ongoing litigation (Reisner, 2023; Kadrey et al. v. Meta Platforms, Inc.). Further, Meta publicly documented their inclusion of `Books3` in the training data for its first generation of Llama models (Touvron et al., 2023a). In summary:

- We discuss a recent probabilistic extraction method (Hayes et al., 2025), which, unlike prior approaches, reveals the risk of extracting a particular piece of memorized training data at generation time (Section 2).
- We connect memorization and extraction risk to U.S. copyright law. Memorized training data that are extractable with higher probability—i.e., that would take very few attempts to extract—may raise different issues for copyright than those that exhibit lower probability. We explain why *average* extraction rates—the typical metric in ML papers—don't provide sufficient information to distinguish between the the two. They can't support *specific* claims about how much an LLM has memorized *specific* works. (Section 3).
- We then develop a simple and precise procedure for the probabilistic extraction of specific memorized text, which we apply to a set of 50 books in `Books3`. Through extensive experiments, we find that while most models don't memorize verbatim most books—either in whole or in part—Llama 3 70B and Llama 3.1 70B memorize some books, like *Harry Potter and the Sorcerer's Stone*, *1984*, and *The Great Gatsby* almost entirely (Section 4).
- Given the degree of memorization we observe for *Harry Potter* and Llama 3.1 70B, we attempt to generate a copy of the entire book. We find that it's trivial to use Llama 3.1 70B and beam search to deterministically reconstruct a close-to-identical copy of the entire text, using only a seed prompt consisting of just the first line of the first chapter (Section 5).
- Finally, we discuss how our results have significant implications for copyright cases, though not ones that unambiguously favor either side. Our results complicate defendants' fair use story, but they also complicates plaintiffs' efforts to bring class action lawsuits (Section 6).

2. Memorization is probabilistic

Training-data extraction and memorization are related concepts, but differ in subtle ways. Following Cooper & Grimmelmann (2024), "[m]ost narrowly, when a user intentionally and successfully prompts a model to generate an output that is an exact or near-exact copy of a piece of training data, that is **extraction**." More broadly, "...when an exact or near-exact copy of a piece of training data can be recon-

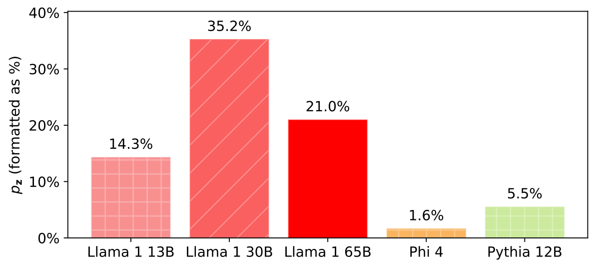


Figure 2: Plotting extraction probability $p_{\boldsymbol{z}}$ for the "careless people" quote from *The Great Gatsby* for different models.

structed by examining the model 'through any means,' that is **memorization**" (Cooper & Grimmelmann, 2024, Part II.A).

As noted in the introduction, it's incorrect to think that extraction is only interesting as a generation-time phenomenon. It's evidence of memorization of training data *inside the model* (Cooper & Grimmelmann, 2024; Carlini, 2025; Schwarzschild et al., 2024; Lee et al., 2023b; Feldman, 2020). In fact, successful extraction is one of the most common forms of evidence in ML for quantifying memorization in LLMs (Nasr et al., 2025; Hayes et al., 2025; Prashanth et al., 2024; Carlini et al., 2023a; Lee et al., 2022; Zhang et al., 2023). Rather than inspecting an LLM's parameters directly, the standard metric in many frontier model-release reports (Gemini Team et al., 2024; Team et al., 2024; Biderman et al., 2023b; Grattafiori et al., 2024) measures **discoverable extraction**: one takes a piece of text from the training data, splits it into a **prefix** and a **target suffix**, uses the prefix as a prompt to the LLM, and counts the example as extracted if the LLM generates text that *exactly* matches the target suffix (Carlini et al., 2021; 2023b; Nasr et al., 2023). For instance, consider the quote from *The Great Gatsby* (Fitzgerald, 1925) in Figure 1. The middle row indicates the verbatim generation of the target suffix in response to the prefix prompt.

Cooper & Grimmelmann (2024) explain why such measurements are incomplete when it comes to the kinds of information relevant to copyright. These measurements capture a simple binary (yes-or-no) outcome of *if* a particular piece of training data was extracted (and thus memorized). But LLMs are probabilistic; they can generate different outputs for the same input—just like in Figure 1. So, instead, a copyright-relevant claim about extraction will likely

> take the following form: a model, when (a) given a particular type of input, will (b) produce a particular type of memorized output, (c) with a particular probability. That probability could be .01 (i.e., a 1% chance), it could be .35 (i.e., a 35% chance) .... The issue for copyright law ... is what to do with this knowledge[:] ... what to do with the fact that element (c)—the probabilistic element—is inescapable (Cooper & Grimmelmann, 2024, Part II.C).

We elaborate on this point in Sections 3 and 6. For now, we note that Hayes et al. (2025) introduce a probabilistic extraction methodology that achieves precisely this. In contrast to the standard *yes-or-no* measurements of discoverable extraction—i.e., seeing if the LLM outputs the verbatim target suffix when prompted with a given prefix—they propose a measure of **probabilistic discoverable extraction**, which quantifies the *probability* (between 0 and 1) that the LLM, under a specific decoding scheme, outputs the verbatim target suffix when prompted with a given prefix. For a training example $\boldsymbol{z}$ that is $a+k$ tokens long, they divide $\boldsymbol{z}$ into an a-length prefix $\boldsymbol{z}_{1:a}$ and a k-length suffix $\boldsymbol{z}_{a+1:a+k}$, and compute the probability $p_{\boldsymbol{z}}$ of extracting example $\boldsymbol{z}$. That is, they compute

$$p_{\boldsymbol{z}} = \prod_{t=a+1}^{a+k} p(z_t \mid \boldsymbol{z}_{1:t-1}), \qquad (1)$$

where the prompt is prefix $\boldsymbol{z}_{1:a}$ and the target suffix is $\boldsymbol{z}_{a+1:a+k}$. This equation just captures that the probability of generating the exact target suffix $\boldsymbol{z}_{a+1:a+k}$ is the product of the probabilities of each token z_t in the suffix, conditioned on all preceding tokens $\boldsymbol{z}_{t-1}$. The overall probability of the suffix is simply the product of these per-token probabilities.[1] As an example, in Figure 2, we plot $p_{\boldsymbol{z}}$ for various models and the "careless people" quote. In practice, it's very simple and efficient to compute conditional per-token probabilities; as a result, $p_{\boldsymbol{z}}$ is also very simple and efficient to compute to quantify verbatim extraction (Appendix A).

In general, even for relatively short suffixes, $p_{\boldsymbol{z}}$ should be very small; it involves repeatedly multiplying together probabilities (i.e., numbers smaller than 1). For example, let's consider that each token z_t in a 50-token suffix has conditional probability 90%—a very high probability, considering that the entire token vocabulary is large (e.g., $32,000$ for LLAMA 1 models). This means all of the other tokens combined (e.g., $31,999$ for LLAMA 1) share the remaining probability of 10%. The probability of this whole suffix would be 0.9 (i.e., 90%) multiplied 50 times: $0.9^{50} \approx 0.005$—just a 0.5% chance that it would be generated verbatim.[2] But in Figure 2, the probability of the target suffix is larger than this *for every model*. For LLAMA 1 30B, $p_{\boldsymbol{z}}$ is over 35%[3]—greater than $1/3$ generations with this prompt will result in the *verbatim* suffix, more often than illustrated in Figure 1!

Such high probabilities *are*, by definition, memorization; they indicate that the training data can be reconstructed through prompting the model with (relatively) few attempts.

[1]For numerical stability, we compute (1) as the $\exp$ of the summed per-token $\log$ probabilities (Appendix A).

[2]Note that, as we make the suffix longer, $p_{\boldsymbol{z}}$ becomes vanishingly small, tending toward 0%. This means that it's very unlikely that we'll be able to extract even a highly memorized long sequence. A 100-token suffix where each token has conditional probability 0.9 has $p_{\boldsymbol{z}} = 0.9^{100} = 0.002\%$; it's $250\times$ less likely than a 50-token sequence with the same per-token probabilities.

[3]This corresponds to an average per-token conditional probability of nearly 98%, i.e., $0.9793^{50} \approx 35.2\%$.

This is precisely what makes memorization interesting. *Any* arbitrary sequence should have low probability, so the fact that certain sequences have high probability—high enough for the model to generate the verbatim suffix with meaningful likelihood—"can't be due to chance" (Carlini, 2025). These probabilities reflect patterns, "statistical correlations" (Concord Music Group, Inc. v. Anthropic PBC) that the model has learned from the training data. But when these probabilities are unusually large, "the pattern *is* the memorized training data", copied inside the model (Cooper & Grimmelmann, 2024, Part II.C).

All of the probabilities in Figure 2 are large; they indicate each model memorized the "careless people" quote in Figure 1. But the quote isn't as easily extractable (i.e., isn't as strongly memorized) for different models. While it would in expectation take fewer than 3 queries (i.e., $1/0.352 \approx 2.84$) with LLAMA 1 30B to extract the quote, it would take over 62 with PHI 4. That is, the different models exhibit different degrees of **extraction risk** (Hayes et al., 2025) for the same example. Both memorization and extraction risk may have implications for copyright, which we discuss in the next section.

3. Memorization, extraction, and copyright

There are dozens of pending copyright suits based on the training and output of LLMs (Chat GPT Is Eating the World). Those copyright disputes generally present three interrelated issues: (1) whether training an LLM on copyrighted material is **fair use** (i.e., limited use of the copyrighted material can, under certain circumstances, be used without permission from the copyright owner); (2) whether the model itself is a **copy** or **derivative work** (Section 6) of the works on which it is trained; and (3) whether the model outputs copyrighted material. Some suits present only one issue, while others present all three questions. Some suits are based on content owned by a single company, while others are class action lawsuits purporting to represent all book authors. There are other copyright issues, as well (Lee et al., 2023b; Samuelson, 2023; Sag, 2024a; Goodyear, 2025; Sobel, 2024; Cooper et al., 2024), for example, whether novel outputs of generative-AI systems are themselves copyrightable (Lee et al., 2023b; United States Copyright Office).

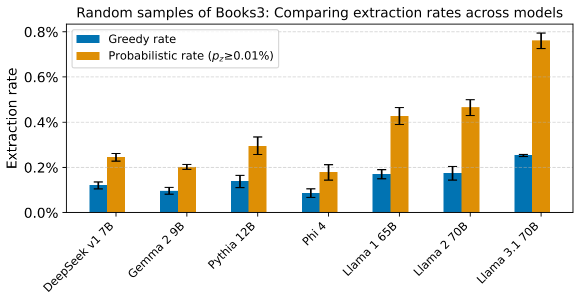
Our paper is not about training and fair use. Those issues have been discussed extensively elsewhere (Lemley & Casey, 2021). Nor is our paper predominantly about the outputs of LLMs and copyright infringement (Cooper et al., 2023; Lee et al., 2023b; Henderson et al., 2023; Lee et al., 2024; Cyphert, 2024; Sag, 2024b; Bracha, 2024), although we use extraction of content from models at generation time as evidence for memorization (Section 2). Our main focus is memorization (Section 2).[4]

Memorization may matter for copyright law in two ways. First, if a model memorizes all or a substantial portion of a copyrighted work (near-)verbatim, the model itself may be an infringing copy or derivative work (Lee et al., 2023b; Cooper & Grimmelmann, 2024). As discussed above (Sections 1 & 2), this is because memorized training data are encoded inside the model. Others have argued that encoding the work in the form of model weights satisfies the technical definition of "copy" in the U.S. Copyright Act (Cooper & Grimmelmann, 2024). That copy, like the copies of copyrighted works used as training data during training, might be fair use, but the analysis of fair use would look somewhat different than *internal* use of works in training. This is particularly true for open-weight models like different DEEPSEEK and LLAMA models, which are not merely used internally by the developer, but are themselves shared externally with others (Section 6). Second, memorization of all or part of a particular work may increase the likelihood that the output of the model may be substantially similar to the copyrighted work. Works that have higher extraction probabilities exhibit a greater degree of extraction risk; it takes fewer attempts to produce an output that is a verbatim copy of such a work (Section 2). That output at generation time will be judged separately from the model (which is also an artifact that is independent of its outputs), and is less likely to be a fair use (Henderson et al., 2023; Lee et al., 2023b).

Two important considerations about memorization and extraction follow from these points: (1) average extraction rates don't provide detailed information about memorization of specific works, and (2) it's a challenging empirical question to draw lines around which probabilities $p_{\boldsymbol{z}}$ (1) may be meaningful for copyright.

Average extraction rates aren't all you need. Numerous studies have shown that models memorize certain amounts of the works they were trained on (e.g., Carlini et al., 2021; Nasr et al., 2023; Hayes et al., 2025; Lee et al., 2022; Biderman et al., 2023b; Gemini Team et al., 2024; Team et al., 2024; Biderman et al., 2023a). In this research, memorization is typically quantified through an overall extraction rate—much like Figure 3 (left). That is, for some (part of a) training dataset, researchers draw (typically at random) examples of a specified length (e.g., 100 tokens), prompt with the first half of the example (i.e., the example prefix), and count extraction as successful if the resulting generation matches the target suffix (Section 2). The extraction rate is computed as the number of attempted extractions that succeeded, relative to the total number of attempts. These reported averages are generally small, just as we observe in Figure 3; this often forms the basis of defendants in copyright infringement suits calling memorization a rare "bug" (Cooper & Grimmelmann, 2024; OpenAI, 2024).

[4] We touch on aspects of LLM outputs (Sections 5 & 6) and will discuss additional nuances in future work.



Model	Result	% of book extracted with $p_z \geq$			
		$\mathbf{75\%}$	$\mathbf{50\%}$	$\mathbf{10\%}$	$\mathbf{1\%}$
LLAMA 3.1 70B	*Harry Potter*	16.75%	43.26%	75.44%	90.89%
	Sandman Slim	0.00%	0.12%	0.28%	0.38%
LLAMA 1 65B	*Harry Potter*	1.69%	4.40%	15.00%	25.48%
	Sandman Slim	0.09%	0.10%	0.13%	0.27%
PYTHIA 12B	*Harry Potter*	0.00%	0.08%	0.10%	0.40%
	Sandman Slim	0.22%	0.26%	0.32%	0.34%

Figure 3: (**left**) Comparing *average* extraction rates of `Books3` text. For different models, we show the traditional (greedy) discoverable extraction rate (blue) and the probabilistic extraction rate (Hayes et al., 2025) (the proportion of examples $\boldsymbol{z}$ for which $p_{\boldsymbol{z}} \geq 0.01\%$, orange) (Appendix E). Regardless of which extraction metric we use, average extraction rates are low. (**right**) A more nuanced story emerges if we examine extraction for specific books. For two books—*Harry Potter* (Rowling, 1998) and *Sandman Slim* (Kadrey, 2009), one of the books by plaintiff Richard Kadrey in *Kadrey et al. v. Meta, Inc.* (Kadrey et al. v. Meta Platforms, Inc.))—we show the proportion of the entire book (in %) that can be extracted. We consider different minimum probabilities of extracting an example $p_{\boldsymbol{z}}$ (75%, 50%, 10%, 1%) and, for each, we compute how much of the total text can be extracted with *at least* that $p_{\boldsymbol{z}}$. Over 90% of *Harry Potter* can be extracted from LLAMA 3.1 70B with $p_{\boldsymbol{z}} \geq 1\%$.

However, while average extraction rates are useful for estimating *overall* memorization of `Books3` text, they clearly tell an incomplete story about the degree to which *specific*, underlying pieces of text have or haven't been memorized by the model. Low extraction rates signal that models likely don't memorize *most* text in `Books3`, but it's possible that *specific* pieces of text are highly memorized. We observed this already with Figure 2 for the "careless people" quote, and we can similarly see this in the Table in Figure 3. With respect to probabilistic extraction (1), some books, like Richard Kadrey's *Sandman Slim* (Kadrey, 2009) are hardly memorized at all.

In contrast, *Harry Potter and the Sorcerer's Stone* (Rowling, 1998) is highly memorized in LLAMA models, especially LLAMA 3 70B and LLAMA 3.1 70B (Section 4). For this model, nearly half of the book can be reconstructed with $p_{\boldsymbol{z}} \geq 50\%$. That is, for over 43% of the text of *Harry Potter*, there exist 50-token prompts where, more than half of the time these prompts are input to the model, the model generates the exact next 50 tokens of the book. In expectation, it takes ≤ 2 prompts (with a given 50-token prefix from the book) to generate the next 50 tokens verbatim. Over 75% of the book can be reconstructed wtih $p_{\boldsymbol{z}} \geq 10\%$ and, similarly, over 90% with $p_{\boldsymbol{z}} \geq 1\%$. While these thresholds correspond to extracting text less reliably, for an LLM, they still reflect enormous probabilities.[5] The results for $p_{\boldsymbol{z}} \geq 1\%$ are strong evidence that LLAMA 3.1 70B effectively has memorized the entire book (Section 5).

Further, using average extraction rates, prior work also observes that larger models exhibit larger amounts of memorization (Carlini et al., 2023b; Hayes et al., 2025). Our results on specific books show that this pattern doesn't cleanly generalize to the extent to which models memorize specific pieces of text. In Figure 2, the smaller LLAMA 1 30B exhibits a significantly higher $p_{\boldsymbol{z}}$ than the larger LLAMA 1 65B—35.2% compared to 21.0%. Similarly, with respect to $p_{\boldsymbol{z}} \geq 75\%$, we estimate that LLAMA 3.1 70B memorizes 0.00% of the text of *Sandman Slim*, while the smaller LLAMA 1 65B and PYTHIA 12B memorize 0.09% and 0.22%, respectively. (We describe how we compute these estimates in Appendix F.)

It's necessary (though nontrivial) to reason about "meaningful" extraction probabilities. We show clear evidence that LLAMA 3.1 70B memorizes almost all of *Harry Potter and the Sorcerer's Stone*. For this model, there exists a prefix/suffix combination such that almost every piece of text in the entire book has a high extraction probability ($p_{\boldsymbol{z}} \geq 1\%$, Figures 3 & 5). These probabilities are too high to be due to random chance; they're clearly memorization (Cooper & Grimmelmann, 2024; Carlini et al., 2021). In contrast, $p_{\boldsymbol{z}}$ being close to 0% indicates that an example isn't memorized (with respect to our specific extraction methodology). This is the case for PYTHIA 12B on nearly every example in *Harry Potter* (Section 4, Figure 5). However, most cases aren't this clear; it's not obvious where to draw a line between the enormous range in between these two extremes.

Put differently, is there a sufficiently small value of $p_{\boldsymbol{z}}$, such that generating the target suffix for $\boldsymbol{z}$ isn't meaningful for copyright—i.e., that it's effectively an instance of "a monkey at the typewriter" (Borel, 1913; The Simpsons: Last Exit to Springfield (Fox television broadcast Mar. 11, 1993))? While we don't address this in this work, we very conservatively only consider sequence extraction probabilities that are so high that they aren't due to happenstance ($p_{\boldsymbol{z}} \geq 0.0001 = 0.01\%$, i.e., average per-token conditional probability $\geq 83\%$). Nevertheless, it's also worth noting

[5] For $p_{\boldsymbol{z}} = 1\%$, $p_{\boldsymbol{z}} \geq 10\%$ and $p_{\boldsymbol{z}} \geq 50\%$ have average per-token conditional probabilities over 91.2%, 95.4%, and 98.6%.

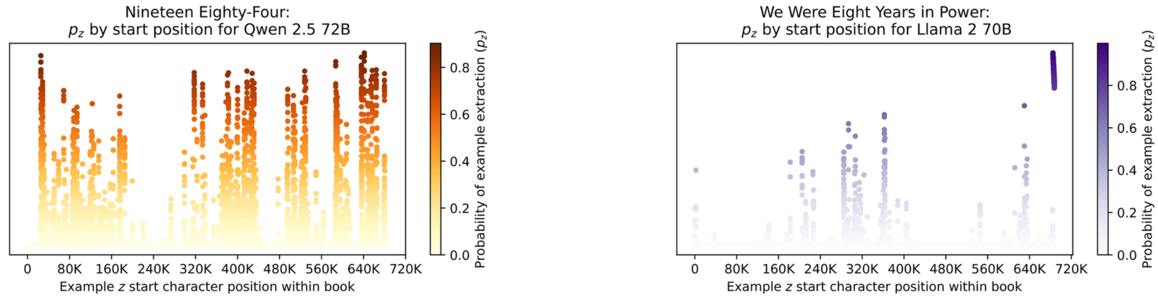


Figure 4: We plot $p_{\boldsymbol{z}}$ for each example according to its start position within the overall book for *1984* (Orwell, 1949) in QWEN 2.5 72B (**left**) and *We Were Eight Years in Power* (Coates, 2017) in LLAMA 2 70B (**right**).

that, as others have observed, LLMs aren't like monkeys randomly outputting tokens (Cooper & Grimmelmann, 2024). The model has learned to generate structured, grammatically correct sentences. That means the actual distribution of possible tokens an LLM would reasonably output is considerably smaller than completely random output would suggest. That possibility, which we hope to explore in further work, bears on the question of how high a $p_{\boldsymbol{z}}$ we should consider to be relevant as evidence of memorization of a specific book rather than, say, learning that in most sentences verbs follow subjects and precede objects.

4. Quantifying memorization for books

We now dig deeper into average extraction rates, showing the presence and extent of memorization in open-weight models for a sample of books drawn text from `Books3`. Overall, we show how memorization varies across models, across books, and within individual books. Memorization for some (but not all) books and for some (but not all) models is surprisingly high. After describing our experimental overall setup, we detail our methodology for identifying memorization "hot-spots" within a book. These experiments help us narrow in on locations within books to explore in further experiments. They also enable us to compare the memorization of different pieces of text within a model, and memorization of a given piece of text across models.

Setup. Similar to Hayes et al. (2025), we use top-k decoding as the sampling algorithm, with temperature $T = 1$ and $k = 40$. The experiments referenced in this section involve 10 continuation-style (i.e., non-chatbot) models: PYTHIA 12B (Biderman et al., 2023b), PHI 4 (a 14B model) (Abdin et al., 2024), LLAMA 1 13B, LLAMA 1 65B (Touvron et al., 2023a), LLAMA 2 70B (Touvron et al., 2023b), LLAMA 3 70B, LLAMA 3.1 70B (Grattafiori et al., 2024), DEEPSEEK V1 67B (DeepSeek-AI et al., 2024), QWEN 2.5 72B (Qwen et al., 2025), and GEMMA 2 27B (Team et al., 2024). PYTHIA was trained on The Pile (Gao et al., 2020) (which contains the torrented `Books3` dataset (Reisner, 2023)), and Touvron et al. (2023a) explicitly note that LLAMA 1 models were trained on `Books3`. PHI 4 was trained predominantly on synthetic data. We describe our choices of models and books to test in Appendix F.1. We ran experiments across `Books3` for computing average extraction rates, but otherwise limit ourselves to a selection of 50 books (out of the nearly $200,000$) that are in `Books3`. These books include those listed with the associated plaintiffs in the (amended) class action complaint of Kadrey et al. v. Meta (Kadrey et al. v. Meta Platforms, Inc., pp. 4-5), as well as generally popular books (e.g., Rowling, 1998; Tolkien, 1937; Camus, 1955; Heller, 1961)), and some academic books (e.g., Zittrain, 2008; Barolini, 2006)). We limit ourselves to only a brief selection of results; the majority can be found in the Appendix.

We suspected that some popular books would exhibit high degrees of memorization (due to duplicated text from other sources), and that more obscure books wouldn't. Overall, our results shouldn't be read as a complete account of memorization across the entire `Books3` dataset. Nor did we select the books at random; we deliberately tried to capture variation across the dataset. We report results for all the books we ran (Appendices D–F). In the future, we'll expand to a much wider set of models and books. In general, we refer to the Appendix, which contains results from thousands of experiments on 17 models. With one exception (Section 5), we ran all experiments on the same 4 A100 GPUs.

Locating memorized book snippets. To identify regions of memorization, we take the following "panning for gold" approach. For a given book, we start at the beginning of the text file in `Books3`. We sample a chunk of text that is sufficiently long to contain 100 tokens of corresponding tokenized text, slide 10 *characters* forward in the book text and repeat this process. We do this for the entire length of the book, which results in approximately one example every 10 characters, e.g., *The Great Gatsby* has $270,870$ characters and thus roughly $27,000$ examples. For most of our experiments, each example is 100 tokens long, which we divide into a 50-token prefix prompt and a 50-token suffix that we attempt to extract.

These 100-token examples overlap significantly; *this is deliberate*. Since we don't often know how open-weight

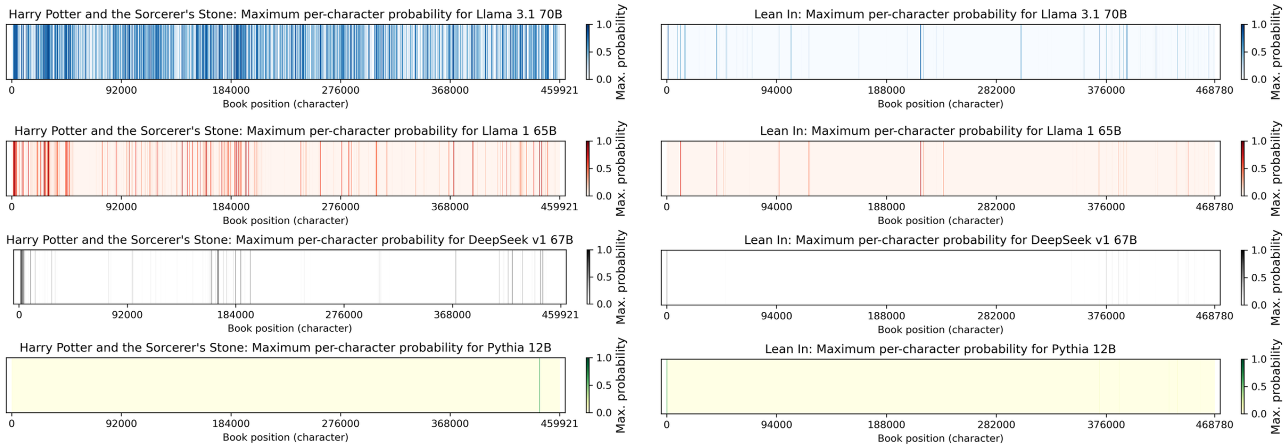


Figure 5: Heatmaps for (top to bottom) LLAMA 3.1 70B, LLAMA 1 65B, DEEPSEEK V1 67B, and PYTHIA 12B for *Harry Potter* (**left**) and *Lean In* (**right**), each showing regions of extraction "hot-spots" according to location (character position) in the book. To highlight these regions, for overlapping examples in the sliding window, we plot the *highest* extraction probability at the particular suffix character location. Each example is 100 tokens (50-token prefix and 50-token suffix). LLAMA 3.1 70B memorized most of *Harry Potter*, reflected in the number of high-probability regions.

models were trained, it's not exactly clear how we should break up books text into examples when attempting extraction. By testing overlapping examples, we expect to surface high-probability *regions* of memorized content within a book (Figure 4, right), which we can then explore more precisely in follow-up experiments, discussed below. We *don't* expect this approach to result in high per-book extraction rates, as this strategy should result in many 0-probability sequences (Appendix F.1). This isn't our goal: we're trying to surface as much total memorization possible. We can then use these memorization "hot-spots" to identify longer memorized sequences (Appendix G).

In Figure 4, we provide results of this "panning for gold" approach for QWEN 2.5 72B for *1984* (Orwell, 1949) and LLAMA 2 70B for *We Were Eight Years in Power* (Coates, 2017). We show each example's extraction probability plotted according to its location (start character position) within the book. For these location plots, white gaps along the x-axis indicate that an example (with that start position) was not extracted (i.e., has 0% probability). A continuous horizontal band of color (associated with the probability on the y-axis) would indicate that we can extract the entire book (with respect to 100-token examples) with that associated probability. Both books show varying amounts of memorization. QWEN 2.5 72B memorizes separate regions of *1984*, and to various degrees of probability; LLAMA 2 70B memorizes fewer and more fragmented pieces of *We Were Eight Years in Power*, and to a lesser degree.

As an alternate view, we also provide heatmaps of extraction probabilities across a book. We show results for *Harry Potter and the Sorcerer's Stone* (Rowling, 1998) and *Lean In* (Sandberg, 2013) (Figure 5). These heatmaps are more condensed than Figure 4: they don't convey how many examples are extractable at different probabilities at each location. However, they facilitate high-level comparisons of memorization "hot-spots" across models at the same book location. At each point on the heatmap, there exists (at least one example) at that location in the book that is extracted with the shown probability. For LLAMA 3.1 70B, there exists an example that can be extracted with high probability at *almost every location in the book*. On it's own, *this doesn't mean (almost) the whole book can be extracted in one continuous segment at generation time; this is a claim about how much of the book we estimate to be memorized in the model's parameters.* We return to reconstructing the entire book at generation time in Section 5.

Comparing memorization across books and models. With these heatmaps, we can also make comparisons about the extent to which different models memorize different books, and how memorization varies across books for a given model. Consider Figure 5. LLAMA 3.1 70B clearly has memorized more of *Harry Potter* (and with higher probability) than the earlier generation, similarly-sized LLAMA 1 65B. For LLAMA 1 65B, there are also contiguous regions (e.g., starting at around character position $190,000$) of highly-memorized text, however they are shorter and fewer in number than those we observe for LLAMA 3.1 70B. Even though both are Meta models, they clearly exhibit important differences in how much they memorize of specific books (Figure 3, right)—not just how much they memorize across `Books3` more generally (Figure 3, left). In contrast, DEEPSEEK V1 67B memorizes significantly less of *Harry Potter* than any LLAMA model of a similar size (Appendix F.1.39). With the same approach, we can only extract short fragments of text, and only a select few at the beginning of the book exhibit high probability. PYTHIA

"Now, yer mum an' dad were as good a witch an' wizard as I ever knew. Head boy an' girl at Hogwarts in their day! Suppose the myst'ry is why You-Know-Who never tried to get 'em on his side before... probably knew they were too close ter Dumbledore ter want anythin' ter do with the Dark Side.

"Maybe he thought he could persuade 'em... maybe he just wanted 'em outta the way. All anyone knows is, he turned up in the village where you was all living, on Halloween ten years ago. You was just a year old. He came ter yer house ~~an'— an'—"~~ **an' — an' —"**

Hagrid suddenly pulled out a very dirty, spotted handkerchief and blew his nose with a sound like a foghorn.

"Sorry," he said. "But it's that sad — knew yer mum an' dad, an' nicer people yeh couldn't find — ~~anyway...~~ **anyway..."**

"You-Know-Who killed 'em. An' then — an' this is the real myst'ry of the thing — he tried to kill you, too. Wanted ter make a clean job of it, I suppose, or maybe he just liked killin' by then. But he couldn't do it. Never wondered how you got that mark on yer forehead? That was no ordinary cut. That's what yeh get when a ~~powerful,~~ **Powerful,** evil curse touches yeh — took care of yer mum an' dad an' yer house, even — but it didn't work on you, an' that's why yer famous, Harry. No one ever lived after he decided ter kill 'em, no one except you, an' he'd killed some o' the best witches an' wizards of the age — the McKinnons, the Bones, the Prewetts — an' you was only a baby, an' you lived."

Something very painful was going on in Harry's mind. As Hagrid's story came to a close, he saw again the blinding flash of green light, more clearly than he had ever remembered it before — and he remembered something else, for the first time in his life: a high, cold, cruel laugh.

Hagrid was watching him sadly.

"Took yeh from the ruined house myself, on Dumbledore's orders. Brought yeh ter this ~~lot..."~~ **lot...."**

Slipping and stumbling, they followed Hagrid down what seemed to be a steep, narrow path. It was so dark on either side of them that Harry thought there must be thick trees there. Nobody spoke much. Neville, the boy who kept losing his toad, sniffed once or twice.

"Yeh'll get yer firs' sight o' Hogwarts in a sec," Hagrid called over his shoulder, "jus' round this bend here."

~~There was a loud "Oooooh!"~~

The narrow path had opened suddenly onto the edge of a great black lake. Perched atop a high mountain on the other side, its windows sparkling in the starry sky, was a vast castle with many turrets and towers.

"No more'n four to a boat!" Hagrid called, pointing to a fleet of little boats sitting in the water by the shore. Harry and Ron were followed into their boat by Neville and Hermione.

"Everyone in?" shouted Hagrid, who had a boat to himself. "Right then — FORWARD!"

And the fleet of little boats moved off all at once, gliding across the lake, which was as smooth as glass. Everyone was silent, staring up at the great castle overhead. It towered over them as they sailed nearer and nearer to the cliff on which it stood.

"Heads down!" yelled Hagrid as the first boats reached the cliff; they all bent their heads and the little boats carried them through a curtain of ivy that hid a wide opening in the cliff face. They were carried along a dark tunnel, which seemed to be taking them right underneath the castle, until they reached a kind of underground harbor, where they clambered out onto rocks and pebbles.

"Oy, you there! Is this your toad?" said Hagrid, who was checking the boats as people climbed out of them.

Figure 6: Two samples of the diff between the ground-truth text of *Harry Potter* (Rowling, 1998) from Books3 and the entire book text we generated using LLAMA 3.1 70B, starting with a single seed prompt of the first line (60 tokens) of chapter 1.

12B is a much smaller model that was also trained on *Harry Potter* (Biderman et al., 2023b; Gao et al., 2020). As noted in Section 3, it effectively memorizes *none* of the book.

Compared to *Harry Potter*, we are able to extract much less memorized training data of *Lean In* from either LLAMA model or DEEPSEEK V1 67B. That is, comparing heatmaps clarifies how the same model can have varying amounts and degrees of memorization for different books. PYTHIA 12B, however, exhibits similar (very low) memorization profiles for both *Lean In* and *Harry Potter*. Together, these heatmaps underscore the importance of how training choices impact memorization: the mere existence of examples within a training dataset and model size don't necessarily imply a particular conclusion (Lee et al., 2023a; 2022; 2023b; 2024).

We defer discussion of other books to Appendix F.1. Aligning with the overall low average extraction rates we observe for models on Books3 (Section 3), we note that most books—regardless of the model we test—exhibit low amounts of memorization. They have heatmaps that resemble the results we show for PYTHIA 12B. This is true for most of the books we tested written by plaintiffs in *Kadrey et al. v. Meta, Inc.* (Kadrey et al. v. Meta Platforms, Inc.)—with important exceptions (e.g., Coates, 2017). Very popular books like *Harry Potter* exhibit higher degrees of memorization for LLAMA models of all sizes and generations, with newer-generation models of a given size class often (but not always) exhibiting higher amounts and degrees of memorization across books.

In general, we find that LLAMA 3 70B and LLAMA 3.1 70B models memorize significantly more than any other model we tested—both in terms of the quantity of examples we can extract from different books and the magnitude of extraction probabilities $p_{\boldsymbol{z}}$. These models memorize significantly more than both QWEN 2.5 72B and DEEPSEEK V1 67B across the 50 books we tested (Appendix F.1). QWEN 2.5 72B tends to contain more verbatim memorization than DEEPSEEK V1 67B. PYTHIA 12B and PHI 4, which was trained predominantly on synthetic data (i.e., is likely to not directly contain Books3), are smaller models; we expect them to memorize less than much larger 70B models. However, PYTHIA 12B and PHI 4 also tend to memorize less than LLAMA 1 13B—a model in the same size class. GEMMA 2 27B is the only model of an intermediate size that we test. We defer additional discussion of our extensive analysis across models and 50 books to the Appendix.

5. Reconstructing *Harry Potter* near-verbatim

With the degree of memorization we observe across the entirety of *Harry Potter* for LLAMA 3.1 70B, we realized it should be possible to reconstruct the entire book near-verbatim, using only a single seed prompt of ground-truth text drawn from the book. We begin with an n-token sequence of ground-truth text as the seed prompt. We use the model to generate the next m tokens. We then remove the first m tokens from the prompt and append the generated m tokens, creating a new n-token prompt for the next generation step. We repeat this process, sliding along the length of the book by prompting with n tokens to produce the next m tokens. With appropriate choices of decoding scheme, n, and m, we reasoned that the model could approximate the entire book with high fidelity.

We ultimately successfully reconstructed the book near-verbatim using a seed prompt of only the first line of chapter 1 (60 tokens), a sliding context window of $n = 3000$ tokens for the prompt, and beam search with 8 beams, generating $m = 50$ tokens at each step. After the first 3000 tokens, the context window progresses beyond the initial seed prompt. All of the subsequent prompts don't contain any ground-truth text drawn from the book; they consist entirely of generated text from prior iterations. The model tended to predict end of sequence (EOS) tokens at the ends of chapters. To get around this, we removed the EOS from the generated m tokens before appending them to the prompt, and replaced them with the tokens for "CHAPTER {c+1}", with {c+1} spelled out (e.g., when $c = 2$, we insert "TWO"). The model then successfully continued to generate text from the book.

As illustrated in Figure 6, differences predominantly involve small inconsistencies in formatting: white space, capitalization, use of underscores (_) to indicate italics, etc. The version of *Harry Potter* in `Books3` uses British spelling (e.g., `"Mum"` instead of `"Mom"`). Very occasionally, when there is only a single line in a paragraph, the model skips that line during generation. Using cosine similarity of the two document TF-IDF vectors, we observe a near-perfect score of 0.9999. TF-IDF is a limited metric, as it treats documents as bags of words and thus fails to capture word order. So, as two additional points of comparison, we compute similarity using greedy longest common subsequence matching. We obtain a word-level similarity of 0.992 and a sentence-level similarity (which is more sensitive to formatting differences) of 0.934.[6]

In short, it was easy to produce a copy of *Harry Potter and the Sorcerer's Stone* that is nearly identical to the original. Further, because beam search is a deterministic decoding algorithm, our results are reproducible. We provide the complete (and very short) code file in Appendix I.

6. Takeaways for memorization and copyright

Our results complicate the traditional narrative both plaintiffs and defendants typically use in copyright cases in describing how LLMs work. The evidence supports the positions of plaintiffs in some respects and of defendants in other respects. More generally, we show that the extent of memorization in models varies with model size, the specific choice of model, the book tested, and even within individual books (Section 3 & 4). We see three primary implications of our results for copyright disputes.

There is definitely some memorization of books in many models. In the case of some models, there's quite a lot of memorization of some books, though most books are not memorized at all (with respect to our specific extraction methodology)—either in whole or in part. Such evidence of memorization in the model matters for the dispute over whether the models themselves are derivative works, an argument that courts have thus far not been receptive to. A work is not a derivative work unless it's "substantially similar" in significant part to the original work (e.g., Litchfield v. Spielberg, 736 F.2d 1352, 1357 (9th Cir. 1984)). And with exceptions (like our *Harry Potter* example), models are not, as plaintiffs sometimes contend, mere copies of the works on which they trained. But our data bolsters plaintiffs' argument that at least some models may be derivative works of at least some books, because the model has memorized a significant amount of protectable expression from the book. The law doesn't require that the entire work be included in the derivative; it's enough that the derivative incorporates a substantial amount of protectable expression. That turns out to be true for some (but not all) books for some (but not all) models

The inquiry doesn't end here; the model, like the training dataset, may be protected under copyright's fair use doctrine (Lemley & Casey, 2021). But the fair use analysis of the model itself may be different than the analysis of the training dataset. A training-data example is an intermediate copy that isn't itself sold, and is only used internally in the course of producing outputs—many of which won't be substantially similar to any training data (Henderson et al., 2023; Lee et al., 2023b). A model, too, is an intermediate work in many cases; its use is to produce outputs, and those outputs are overwhelmingly not copyright infringements. But for companies that sell or release their models to others under open source licenses, the model itself is the product, and sometimes one that is being sold directly for commercial gain. That may make it harder to rely on the cases that justify training as fair use (e.g., Author's Guild v. Google, Inc., 804 F.3d 202 (2d Cir. 2015); Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992), as amended (Jan. 6, 1993); Sony Computer Ent., Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000)) (Section 3).

The consequences of a finding that the model itself is a copy of some copyrighted works, and that a **distribution** of the model was thus a distribution of a copy of those works, could be dramatic for the AI industry. LLAMA 3.1 70B was downloaded 105,029 times in May 2025 on HuggingFace (Appendix, Figure 10)—far less from the height of its release in 2024. If we say conservatively that the model has been downloaded 1 million times since its release, then those 1 million downloads could be seen as 1 million potentially infringing distributions of reproductions of *Harry Potter and the Sorcerer's Stone*—as well as any other books in copyright for which a more than *de minimis* amount of copyrighted expression has been memorized. With $150,000 (potentially) per infringing work, even if 3% of `Books3` were to be found to be infringing copies in the model (something that, to be clear, we have *not* studied), that's a potential maximum statutory damages award of nearly $1 billion dollars. If the model itself is a copy and isn't protected by fair use, courts might also order that it not be distributed or even that copies of the model itself be destroyed (Copyright Law of the United States, 2010). It's not an exaggeration to say that the risk of damages across all memorized books and of orders of destruction are an existential threat to the companies training these models.

The extent of memorization varies in several important ways. Our experiments show that the extent of memorization varies widely from model to model and, within a model,

[6] We apply minimal text normalization before computing these values: removing _ from both documents (used for italics in the `Books3` version) and aligning ellipses to `"..."` (which appear as `" .   .   . "` in `Books3`).

even from work to work in the Books3 dataset. This means it's hard to make any sort of **class**-wide (in the class-action-lawsuit sense) general assessment of whether a particular model copied a particular work and whether, for that model, infringing output based on memorization is even possible. Indeed, we show that memorization rates vary for many of the actual named authors in lawsuits, and even from book to book for the same plaintiffs (e.g., Ta-Nehisi Coates, see Appendix F). Because many of the pending cases are proceeding as class actions, plaintiffs will have to demonstrate that *all* book owners have sufficient common legal and fact issues, such that it makes sense for a court to certify the class and treat them all together (Fed. R. Civ. P. 23(a)(2)).

That may be impossible because, as is clear from our results, basic questions about whether the model actually incorporated any significant expression from any individual plaintiff's book can't be generalized. Some plaintiffs may be able to show copying, but others won't. Courts generally deny class certification in such circumstances (e.g., Wal-Mart Stores, Inc. v. Dukes, 564 U.S. 338, 349–50 (2011)). To be certified as a class action, the plaintiffs generally have to show that all putative class members not only raise common legal issues but that they share common injuries, as well. Our evidence suggests that will be difficult to do without running every book through a test similar to the ones we performed here.

There's no deterministic path from model memorization to outputs of infringing works. While we've used probabilistic extraction as proof of memorization, to actually extract a given piece of 50 tokens of copied text often takes hundreds or thousands of prompts. Using the extraction method of Hayes et al. (2025), we've proven that it can be done, and therefore that there is memorization in the model (Cooper & Grimmelmann, 2024; Carlini et al., 2021). But this is where, even though extraction is evidence of memorization, it may become important that they are not identical processes (Section 2). Memorization is a property of the model itself; extraction comes into play when someone *uses* the model (Cooper & Grimmelmann, 2024). This paper generally makes claims about the former, not the latter.

Nevertheless, it's worth mentioning that, in the average case, it's unlikely anyone in the real world would actually use the model in practice with this extraction method to deliberately produce infringing outputs, because doing so would require huge numbers of generations to get non-trivial amounts of text in practice (Appendix A.1). For the majority of cases we've observed—where models don't exhibit extreme degrees of memorization of examples (e.g., $p_{\mathbf{z}} < 10\%$)—that makes output infringement much less of a real-world problem, even for models that exhibit a lot of memorization overall. This also may affect who courts view as directly liable for any output infringement; someone (like some plaintiffs' lawyers) who runs hundreds or thousands of queries to try to get one that is infringing seems much more like the direct volitional actor misusing the model (CoStar Grp., Inc. v. LoopNet, Inc., 373 F.3d 544 (4th Cir. 2004)). On the other hand, in several cases (e.g., *1984*), it doesn't take thousands or even hundreds of generations to get large amounts of text; it takes only a handful. And in one that we have identified so far (e.g., *Harry Potter and the Sorcerer's Stone*), it's even possible to generate the book directly from a short starting prompt. We'll address this further in future work.

The fact that extraction is probabilistic and difficult to do also matters to the "is training infringement" issue—though it isn't determinative, because it means use of the training data to create **superseding copies** (i.e., a copy that can stand in for/ replace the work) is often unlikely (though possible, see Section 5) in practice. Nonetheless, the fact of memorization creates a point of distinction from *Author's Guild v. Google* (Author's Guild v. Google, Inc., 804 F.3d 202 (2d Cir. 2015)), which considered and rejected an argument that hackers could access the plaintiffs' works through Google's internal database (804 F.3d, at 227-28). Our results show that it's possible to recreate some of the content of some books by "hacking" the model itself. We think it unlikely as a practical matter that people will use the model in this way; there are easier and more effective ways to pirate a book. But at a minimum, it complicates the copyright fair use analysis.

7. Conclusion

Building on the recent work of Hayes et al. (2025)—a novel probabilistic extraction method—we use the Books3 dataset to show that the extent of verbatim memorization of books in open-weight LLMs is more significant than previously described. We also show that memorization varies widely from model to model and from book to book within each model, as well as varying in different parts of individual books. While we find that, with respect to probabilistic extraction most models don't memorize most books, this isn't always the case. In extreme instances, a model may memorize a book so significantly that it's possible to generate the entire book near-verbatim, as we've done with LLAMA 3.1 70B for *Harry Potter*. Altogether, our results complicate current disputes over copyright infringement, both by rejecting easy claims made by both sides about how models work and by demonstrating that there's no single answer to the question of how much a model memorizes. There is much more work to do; we only ran book-specific, verbatim-extraction experiments for 50 books in Books3. In future work, we intend to explore non-verbatim extraction, as well as possible explanations for memorization variability and for the high level of memorization of some specific works by particular models.

Impact Statement

Our work has significant implications for current debates about memorization and copyright, with respect to LLMs. In particular, we add nuance to existing debates by re-envisioning how memorization could be measured to have relevance for copyright. Our work further clarifies points first raised in Lee et al. (2023b): it isn't possible to make sweeping statements about all of generative AI (or even all LLMs) and copyright. Models and systems differ significantly; the amount they memorize within the same book can vary significantly, as well as between books. Even if memorization is rare overall, the underlying amounts of memorization for specific books tell a very different story. All of this is relevant for copyright. Developing knowledge in this area can help us build and use models more responsibly.

References

804 F.3d.

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., and Zhang, Y. Phi-4 Technical Report, 2024. URL https://arxiv.org/abs/2412.08905.

amongglue/books3-subset-raw, 2025. URL https://huggingface.co/datasets/amongglue/books3-subset-raw.

Anthropic. Introducing 100K Context Windows, May 2023. URL https://www.anthropic.com/index/100k-context-windows/.

Atwood, M. *The Handmaid's Tale*. McClelland & Stewart, 1985.

Author's Guild v. Google, Inc., 804 F.3d 202 (2d Cir. 2015).

Barolini, T. *Dante and the Origins of Italian Literary Culture*. Fordham University Press, 2006.

Biderman, S., Prashanth, U. S., Sutawika, L., Schoelkopf, H., Anthony, Q., Purohit, S., and Raff, E. Emergent and Predictable Memorization in Large Language Models, 2023a. URL https://arxiv.org/abs/2304.11158.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023b.

Borel, E. La mecanique statique et l'irreversibilite. *J. Phys. Theor. Appl.*, 3(1):189–196, 1913. doi: 10.1051/jphystap:019130030018900. URL https://doi.org/10.1051/jphystap:019130030018900.

Bracha, O. The Work of Copyright in the Age of Machine Production, January 2024. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4581738.

Brown, D. *The Da Vinci Code*. Doubleday, 2003.

Camus, A. *The Myth of Sisyphus*. Hamish Hamilton Publishers, Trinity Press, 1955. Justin O'Brien (translator).

CANBERT/pile_books3_text, 2025. URL https://huggingface.co/datasets/CANBERT/pile_books3_text.

Carlini, N. What my privacy papers (don't) have to say about copyright and generative AI, 2025. URL https://nicholas.carlini.com/writing/2025/privacy-copyright-and-generative-models.html.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. Extracting Training Data from Diffusion Models, 2023a.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. Quantifying Memorization Across Neural Language Models. In *International Conference on Learning Representations*, 2023b.

Carroll, L. *Alice's Adventures in Wonderland*. Macmillan, 1865.

Chat GPT Is Eating the World, 2024. URL https://chatgptiseatingtheworld.com.

Coates, T.-N. *The Beautiful Struggle*. One World, 2009.

Coates, T.-N. *We Were Eight Years in Power: An American Tragedy*. One World, 2017.

Coates, T.-N. *The Water Dancer: A Novel*. One World, 2019.

Cogburn, J. *Dungeons and Dragons and Philosophy*. Open Court Publishing, 2012.

Concord Music Group, Inc. v. Anthropic PBC. 3:23-cv-01092 (M.D. Tenn.).

Cooper, A. F. and Grimmelmann, J. The Files are in the Computer: Copyright, Memorization, and Generative AI. *arXiv preprint arXiv:2404.12590*, 2024.

Cooper, A. F., Lee, K., Grimmelmann, J., Ippolito, D., Callison-Burch, C., Choquette-Choo, C. A., Mireshghallah, N., Brundage, M., Mimno, D., Choksi, M. Z., Balkin, J. M., Carlini, N., Sa, C. D., Frankle, J., Ganguli, D., Gipson, B., Guadamuz, A., Harris, S. L., Jacobs, A. Z., Joh, E., Kamath, G., Lemley, M., Matthews, C., McLeavey, C., McSherry, C., Nasr, M., Ohm, P., Roberts, A., Rubin, T., Samuelson, P., Schubert, L., Vaccaro, K., Villa, L., Wu, F., and Zeide, E. Report of the 1st Workshop on Generative AI and Law. *arXiv preprint arXiv:2311.06477*, 2023.

Cooper, A. F., Choquette-Choo, C. A., Bogen, M., Jagielski, M., Filippova, K., Liu, K. Z., Chouldechova, A., Hayes, J., Huang, Y., Mireshghallah, N., Shumailov, I., Triantafillou, E., Kairouz, P., Mitchell, N., Liang, P., Ho, D. E., Choi, Y., Koyejo, S., Delgado, F., Grimmelmann, J., Shmatikov, V., Sa, C. D., Barocas, S., Cyphert, A., Lemley, M., danah boyd, Vaughan, J. W., Brundage, M., Bau, D., Neel, S., Jacobs, A. Z., Terzis, A., Wallach, H., Papernot, N., and Lee, K. Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy, Research, and Practice. *arXiv preprint arXiv:2412.06966*, 2024.

Copyright Law of the United States. 17 U.S. Code § 503 - Remedies for infringement: Impounding and disposition of infringing articles, December 2010. URL https://www.law.cornell.edu/uscode/text/17/503.

CoStar Grp., Inc. v. LoopNet, Inc., 373 F.3d 544 (4th Cir. 2004).

Cyphert, A. B. Generative AI, Plagiarism, and Copyright Infringement in Legal Documents. *Minnesota Journal of Law, Science & Technology*, 25, 2024.

DeepSeek-AI et al. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism, 2024. URL https://arxiv.org/abs/2401.02954.

Doctorow, C. *Down and Out in the Magic Kingdom*. Tor Books, 2003.

Duffy, C. A. *The World's Wife*. Farrar, Straus and Giroux, 2001.

Díaz, J. *Drown*. Riverhead Books, 1996.

Díaz, J. *The Brief Wondrous Life of Oscar Wao*. Riverhead Books, 2007.

Díaz, J. *This Is How You Lose Her*. Riverhead Books, 2012.

Egan, J. *A Visit from the Goon Squad*. Alfred A. Knopf, 2010.

EleutherAI/the_pile_deduplicated, 2025. URL https://huggingface.co/datasets/EleutherAI/the_pile_deduplicated.

Farnsworth, C. *The President's Vampire*. G.P. Putnam's Sons, 2011.

Fed. R. Civ. P. 23(a)(2). (requiring questions of law or fact common to the class before it can be certified).

Feldman, V. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pp. 954–959, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794.

Fitzgerald, F. S. *The Great Gatsby*. Charles Scribner's Sons, 1925.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gemini Team, Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Gladwell, M. *Blink: The Power of Thinking Without Thinking*. Little, Brown and Company, 2005.

Golden, C. *Dead Ringer*. St. Martin's Press, 2016.

Golden, C. *Ararat*. St. Martin's Press, 2017.

Goodyear, M. P. Artificial Infringement. In *Proceedings of the 2025 Symposium on Computer Science and Law*, CSLAW '25, pp. 26–38, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714214. doi: 10.1145/3709025.3712208. URL https://doi.org/10.1145/3709025.3712208.

Grattafiori, A. et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783.

Greer, A. S. *The Confessions of Max Tivoli*. Picador, 2005.

Grisham, J. *Theodore Boone: The Fugitive*. E.P. Dutton, 2015.

Haddon, M. *The Curious Incident of the Dog in the Night-Time*. Doubleday, 2003.

Hayes, J., Swanberg, M., Chaudhari, H., Yona, I., Shumailov, I., Nasr, M., Choquette-Choo, C. A., Lee, K., and Cooper, A. F. Measuring memorization in language models via probabilistic extraction. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9266–9291, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.469/.

Heller, J. *Catch-22*. Simon & Schuster, 1961.

Henderson, P., Li, X., Jurafsky, D., Hashimoto, T., Lemley, M. A., and Liang, P. Foundation Models and Fair Use, 2023. URL https://arxiv.org/abs/2303.15715.

Hwang, D. H. *M. Butterfly*. Plume, 1988.

Jacobs, B. E. *All the Onions*. Storey Publishing, LLC, 1999.

Joyce, J. *Ulysses*. Paris-Shakespeare, 1922.

Kadrey, R. *Sandman Slim: A Novel*. Harper Collins, 2009.

Kadrey et al. v. Meta Platforms, Inc. URL https://www.courthousenews.com/wp-content/uploads/2025/02/kadrey-vs-meta-third-amended-complaint.pdf. No. 3:23-cv-03417-VC (Third Amended Consolidated Complaint).

Kadrey v. Meta Platforms, Inc. No. 3:23-cv-03417 (N.D. Cal. July 7, 2023).

Klam, M. *Who Is Rich?* Random House, 2017.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pp. 8424–8445, 2022.

Lee, K., Cooper, A. F., Grimmelmann, J., and Ippolito, D. AI and Law: The Next Generation. *SSRN*, 2023a. http://dx.doi.org/10.2139/ssrn.4580739.

Lee, K., Cooper, A. F., and Grimmelmann, J. Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain. *arXiv preprint arXiv:2309.08133*, 2023b.

Lee, K., Cooper, A. F., and Grimmelmann, J. Talkin' 'Bout AI Generation: Copyright and the Generative-AI Supply Chain (The Short Version). In *Proceedings of the Symposium on Computer Science and Law*, CSLAW '24, pp. 48–63, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703331. doi: 10.1145/3614407.3643696. URL https://doi.org/10.1145/3614407.3643696.

Lemley, M. and Casey, B. Fair Learning. *Texas Law Review*, 99:743, 2021.

Lippman, L. *After I'm Gone: A Novel*. Harper Collins, 2014.

Lippman, L. *Sunburn*. William Morrow, 2018.

Litchfield v. Spielberg, 736 F.2d 1352, 1357 (9th Cir. 1984). ("a work is not derivative unless it has been substantially copied from the prior work.").

Ma, I., Domingo, I., Krone-Martins, A., Baldi, P., and Lopes, C. V. Memorization: A Close Look at Books, 2025. URL https://arxiv.org/abs/2504.12549.

Martin, G. R. *A Game of Thrones*. Voyager Books, 1996.

Meyer, S. *Twilight*. Little, Brown and Company, 2005.

Morrison, T. *Beloved*. Alfred A. Knopf Inc., 1987.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable Extraction of Training Data from (Production) Language Models. *arXiv preprint arXiv:2311.17035*, 2023.

Nasr, M., Rando, J., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Tramèr, F., and Lee, K. Scalable Extraction of Training Data from Aligned, Production Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vjel3nWP2a.

Ogawa, Y. *The Memory Police*. Patheon, 2019. Stephen Snyder (translator).

OpenAI. OpenAI and journalism, January 2024. URL https://openai.com/blog/openai-and-journalism.

Orwell, G. *Nineteen-Eighty Four*. Harcourt, Brace and Company, 1949.

Prashanth, U. S., Deng, A., O'Brien, K., au2, J. S. V., Khan, M. A., Borkar, J., Choquette-Choo, C. A., Fuehne, J. R., Biderman, S., Ke, T., Lee, K., and Saphra, N. Recite, Reconstruct, Recollect: Memorization in LMs as a Multifaceted Phenomenon, 2024. URL https://arxiv.org/abs/2406.17746.

Pullman, P. *The Subtle Knife*. Penguin Random House, 1997.

Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 Technical Report, 2025. URL https://arxiv.org/abs/2412.15115.

Reisner, A. These 183,000 Books Are Fueling the Biggest Fight in Publishing and Tech. *The Atlantic*, September 2023. URL https://www.theatlantic.com/technology/archive/2023/09/books3-database-generative-ai-training-copyright-infringement/675363/.

Rouighi, R. *The Making of a Mediterranean Emirate*. University of Pennsylvania Press, 2011.

Rowling, J. *Harry Potter and the Sorcerer's Stone*. Scholastic, 1998.

Rowling, J. *Harry Potter and the Goblet of Fire*. Scholastic, 2000.

Sag, M. Copyright Safety for Generative AI. *Houston Law Review*, 61, 2024a.

Sag, M. Fairness and Fair Use in Generative AI. *Fordham Law Review*, 92, 2024b.

Salinger, J. *The Catcher in the Rye*. Little, Brown and Company, 1951.

Samuelson, P. Generative AI meets copyright. *Science*, 381 (6654):158–161, 2023.

Sandberg, S. *Lean In*. Alfred A. Knopf, 2013.

SaylorTwift/the_pile_books3_minus_gutenberg, 2025. URL https://huggingface.co/datasets/SaylorTwift/the_pile_books3_minus_gutenberg.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z. C., and Kolter, J. Z. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992), as amended (Jan. 6, 1993).

Shelley, M. *Frankenstein*. Lackington, Hughes, Harding, Mavor, & Jones, 1818.

Silverman, S. *The Bedwetter*. Harper Collins, 2010.

Snyder, R. L. *No Visible Bruises*. Bloomsbury Publishing, 2019.

Sobel, B. Elements of Style: Copyright, Similarity, and Generative AI. *Harvard Journal of Law & Technology*, 38, 2024.

Sony Computer Ent., Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000).

Team, G. et al. Gemma 2: Improving Open Language Models at a Practical Size, 2024. URL https://arxiv.org/abs/2408.00118.

TerKeurst, L. *Unglued: Making Wise Choices in the Midst of Raw Emotions*. Zondervan, 2012.

TerKeurst, L. *Embraced*. Thomas Nelson, 2018.

The Simpsons: Last Exit to Springfield (Fox television broadcast Mar. 11, 1993), 1993. ("This is a thousand monkeys working at a thousand typewriters. Soon they'll have written the greatest novel known to man . . . [l]et's see. 'It was the best of times, it was the blurst of times?' You stupid monkey!").

Tolkien, J. *The Hobbit*. George Allen and Unwin, 1937.

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. URL https://arxiv.org/abs/2302.13971.

Touvron, H. et al. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023b. URL https://arxiv.org/abs/2307.09288.

United States Copyright Office. Copyright and Artificial Intelligence (Part 2: Copyrightability), 2025. URL https://www.copyright.gov/ai/Copyright-and-Artificial-Intelligence-Part-2-Copyrightability-Report.pdf.

Wal-Mart Stores, Inc. v. Dukes, 564 U.S. 338, 349–50 (2011). (holding that class certification was inappropriate because commonality requires that class members be capable of showing that they have suffered the same injury).

Wilf-Townsend, D. The Deletion Remedy. *North Carolina Law Review*, 103, 2024. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4933011. Forthcoming 2025.

Woodson, J. *Brown Girl Dreaming*. Penguin Group, 2014.

Woodson, J. *Another Brooklyn*. Amistad Press, 2016.

Woolf, V. *Orlando*. Hogarth Press, 1928.

Zhang, C., Ippolito, D., Lee, K., Jagielski, M., Tramèr, F., and Carlini, N. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362, 2023.

Zittrain, J. *The Future of the Internet and How to Stop It*. Yale University Press, 2008.

Appendix Table of Contents

A. Additional notes on memorization and extraction

We provide additional notes on the background material in this paper, including the "careless people" quote (Appendix A.1), more details on the key metric in this paper ((n, p)-discoverable extraction, Appendix A.2), as well as our implementation (roughly speaking) for this metric (Appendix A.3). Please also refer to Section 2 of the main paper text.

A.1. Additional notes on the "careless people" example

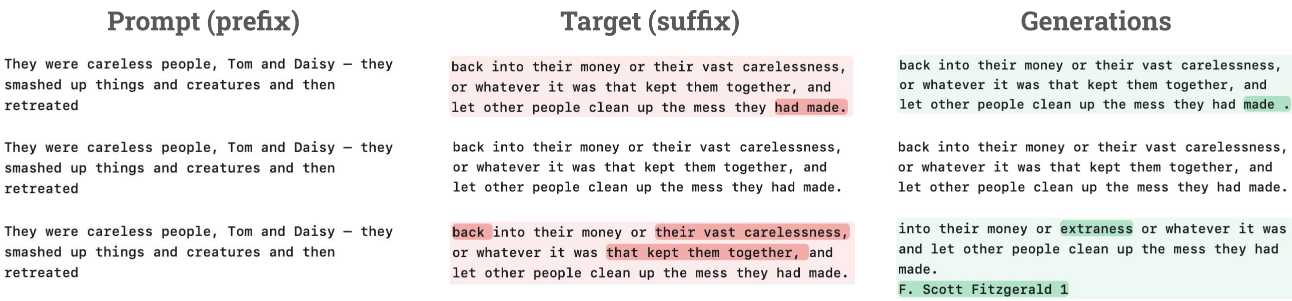
The headline figure we include in Section 2 is meant to be illustrative and catchy—an example, not a general statement about memorization and extraction. We repeat this example in Figure 7a. The example is drawn from *The Great Gatsby*; it's one of the most famous quotes from the book. It's also a very short example (with the LLAMA 1 tokenizer, it is 57 tokens in total). We don't focus on such short suffixes in our analysis, for reasons that are discussed both in the main text and in our experimental details below. (In this case we also clip the prefix length to be quite short.) In the background section, we limit ourselves to something pithy.

It's also the case that very famous quotes like this are duplicated in many locations in training-data sources (e.g., blogs, essays, personal web pages, etc.) We aren't claiming (here, or in general) that this sequence can be extracted *because* a given book (here, *The Great Gatsby*) was included in `Books3`; it could be extracted (or more extractable) because it is duplicated in many places in the training data. Of course, one of those copies is in `Books3`, if the entirety of `Books3` is included in the model's training data. (And we have reason to believe that it was for the LLAMA 1 family (Touvron et al., 2023a): "We include two book corpora in our training dataset: the Gutenberg Project, which contains books that are in the public domain, and the Books3 section of The Pile (Gao et al., 2020)(Gao et al., 2020), a publicly available dataset for training large language models," (p. 2).)

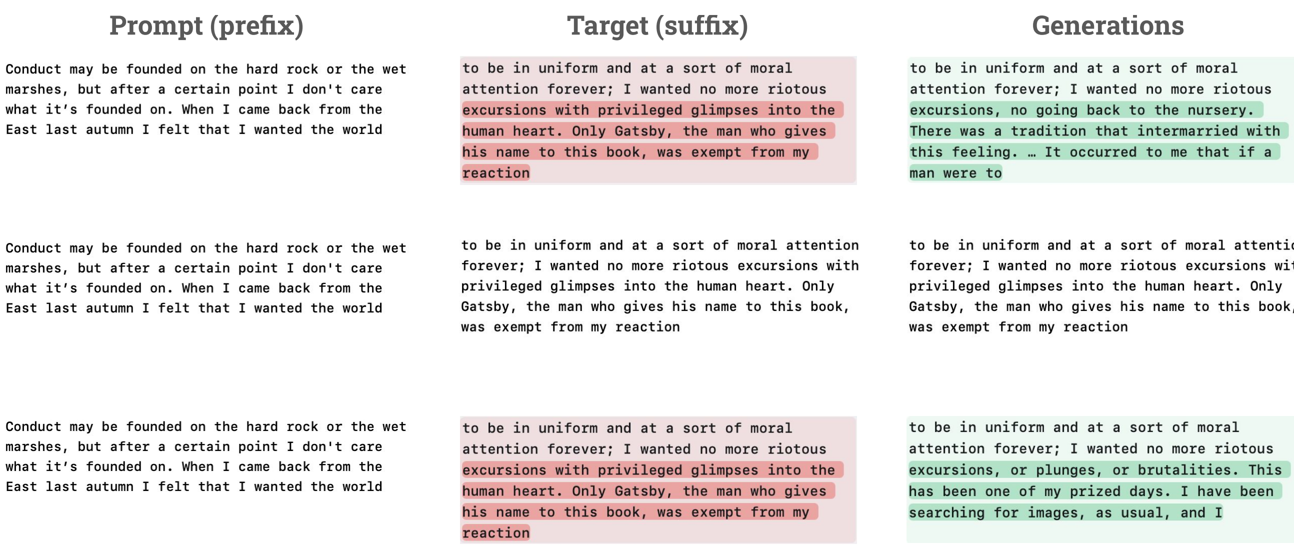
We also include a second example in Figure 7b, which is on-par with the length of examples that we extract in the majority of our experiments (100 tokens in our sliding window and average-extraction-rate experiments). This example was chosen at random from sampling from *The Great Gatsby*, and ensuring that the example began at the beginning of a sentence. (In this case, we sampled a 100-token sequence, and then backed up to the beginning of the sequence, which is what accounts for the additional 7 tokens; the suffix start index was also sampled at random, from indexes 50 through 60 of the full example.) Note, for LLAMA 1 13B (the model we used for these examples), the probability of extracting the second example is much higher than for the famous quote we include in Section 2).

For our verbatim extraction experiments, we compute sequence probabilities directly from the logits we obtain from running sequences through the LLM. We do this rather than generating multiple suffixes for the different prefixes. **These two approaches are functionally equivalent: the computed sequence probabilities correspond to the statistically expected frequencies of verbatim extracted outputs we would observe if we were to generate a large number of sequences for the same prompt.** Hayes et al. (2025), the authors of this extraction approach, confirm this with ample experimental evidence.

In future work, we intend to generate (not just process logits to compute probabilities), as this is required for computing *non*-verbatim extraction metrics. We limit ourselves to verbatim extraction in this paper; it's significantly computationally cheaper to do so, as operating on logits (as opposed to producing multiple generations per prompt) requires only one forward pass through the model. (See Section 2 and Appendix A for more details.) As a result, our work here (in several respects) only scratches the surface: there are various different ways to instantiate this extraction methodology in practice, we only explore a limited set of open-weight models, and we only test low-level extraction attacks on 50 of the nearly $200,000$ books in `Books3`.



(a) 14.3% probability of extracting the verbatim target suffix from LLAMA 1 13B using top-k sampling (prefix: 25 tokens; suffix: 32 tokens; sampling configuration: $k = 40$, $T = 1$).



(b) 57.97% probability of extracting the verbatim target suffix from LLAMA 13B using top-k sampling (prefix: 51 tokens; suffix: 56 tokens; sampling configuration: $k = 40$, $T = 1$).

Figure 7: Two examples of issuing a given prompt (left column) from *The Great Gatsby* to LLAMA 1 13B three times. In each figure, the middle row shows an instance of verbatim extraction; the top and bottom rows show instances where the generations do not match the target suffix.

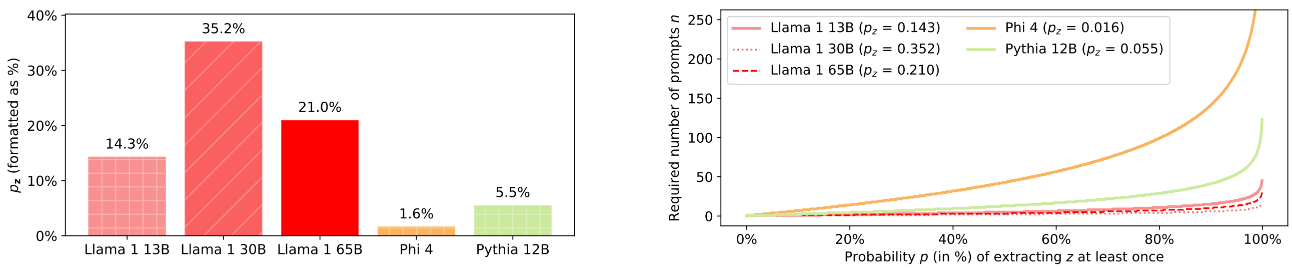


Figure 8: Plotting extraction probability $p_{\boldsymbol{z}}$ for the "careless people" quote from *The Great Gatsby* (see Section 2). (**left**) $p_{\boldsymbol{z}}$ (1) for different models. (**right**) Translating $p_{\boldsymbol{z}}$ into how many prompts n it would take to extract the example z with probability p (2). See Appendix A.1 for details.

A.2. Metrics

Hayes et al. (2025) provide a more formal definition for the metric we compute in this paper, which the authors frame with a different intuition. Beyond directly comparing $p_{\boldsymbol{z}}$ values—for the same example across LLMs, across multiple examples for the same LLM, etc.—Hayes et al. (2025) offer an intuitive alternative. Once one has computed $p_{\boldsymbol{z}}$, with some simple math they can determine how many times n would be necessary to prompt the model with the prefix, in order to guarantee that with probability p that the LLM outputs the exact target suffix. (This is where their metric derives its name, $\boldsymbol{(n,p)}$**-discoverable extraction**.)

Definition 1 (**(n, p)-discoverable extraction**, from Hayes et al. (2025))**.** Given a training example $\boldsymbol{z}$ that is split into an a-length prefix $\boldsymbol{z}_{1:a}$ and a k-length suffix $\boldsymbol{z}_{a+1:a+k}$, $\boldsymbol{z}$ is (n, p)-*discoverably extractable* if

$$\Pr\Big(\cup_{w\in[n]}\,(g_\phi\circ f_\theta)^k_w(\boldsymbol{z}_{1:a}) = \boldsymbol{z}_{1:a+k}\Big)\geq p,$$

where $(g_\phi\circ f_\theta)^k_w(\boldsymbol{z}_{1:a})$ represents the w-th (of n) independent execution of the autoregressive process of generating a distribution over the token vocabulary, sampling a token from this distribution, and adding the token to the sequence $k>0$ times, starting from the same initial sequence $\boldsymbol{z}_{1:a}$.

Note that this definition can be easily adapted to non-verbatim extraction by testing if a generation is within ϵ distance from the target suffix for a given distance metric, rather than testsing for equality with $\boldsymbol{z}_{1:a+k}$. Also note that, in practice for verbatim extraction, computing this metric involves calculating sequence probabilities $p_{\boldsymbol{z}}$ as follows (rather than issuing n independent prompts to the model):

$$1-(1-p_{\boldsymbol{z}})^n\geq p \quad\Rightarrow\quad n\geq\frac{\log(1-p)}{\log(1-p_{\boldsymbol{z}})},\quad \text{where}\;\; p_{\boldsymbol{z}}=\exp\Big(\sum_{t=a+1}^{a+k}\log p(z_t\mid \boldsymbol{z}_{1:t-1})\Big) \tag{2}$$

We compute $p_{\boldsymbol{z}}$ as the $\exp$ of the sum of conditional $\log$ probabilities of the token sequences, as this is more numerically stable that multiplying together the conditional probabilities of the token sequences. (See Section 2.) We show examples of $p_{\boldsymbol{z}}$ for the "careless people" quote in Figure 8 (left), which is a reprint of the same figure in Section 2.

For a given $p_{\boldsymbol{z}}$, one can pick a probability threshold p and get the corresponding number of prompts n (or vice versa), as shown above. Figure 8 (right) visualizes how n changes for different settings of p for the "careless people" quote. Since $p_{\boldsymbol{z}}$ is the probability of generating the suffix with 1 prompt to the model, the probability of *not* generating it in 1 prompt is $(1-p_{\boldsymbol{z}})$. The probability of *not* generating it in n independent prompts to the model (with the same prompt) is $(1-p_{\boldsymbol{z}})^n$, and so the probability of generating it in n prompts is 1 minus this probability, i.e., $1-(1-p_{\boldsymbol{z}})^n$.

As an intuition, think about a fair coin flip coming up heads. Here, $p_{\boldsymbol{z}}=0.5$ (the probability of heads in one flip). With 2 actual flips ($n=2$), the probability of flipping heads *at least once* is $1-(1-0.5)^2=0.75=p$; when $n=10$, $p=0.9990$. We show this intuition in Figure 9 (left), and for LLAMA 1 30B and the "careless people" quote, for which $p_{\boldsymbol{z}}=35.2\%$ (right). In practice, for LLMs, we can compute $p_{\boldsymbol{z}}$ *without* prompting n times, which we discuss further below (Appendix A.3).

In our work, we re-frame this definition to emphasize the quantity $p_{\boldsymbol{z}}$ that we actually compute: the probability of extracting a sequence $\boldsymbol{z}$ (for the given LLM, hyperparameter-configured decoding scheme, and suffix start index location). This is

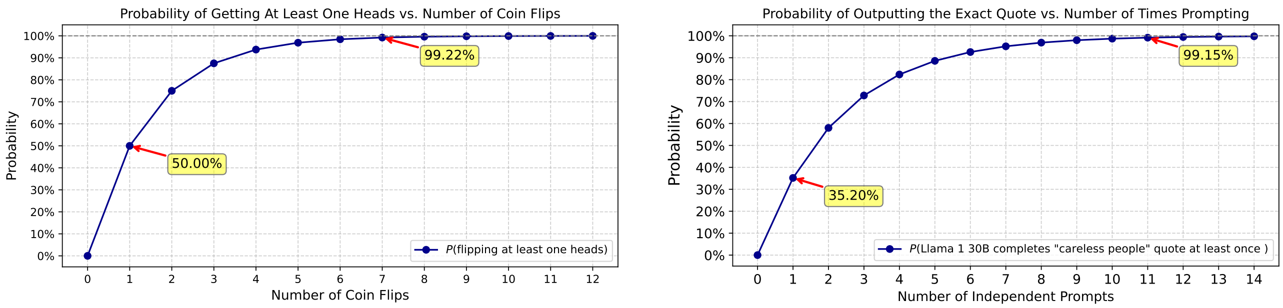


Figure 9: Showing how the probability p of generating a targeted outcome *at least once* changes as a function of n independent trials. Following the intuition of flipping a fair coin (where heads has $p_{\boldsymbol{z}} = 0.5 = 50\%$), we show how the probability of flipping heads at least once changes with more flips (**left**). We show how the probability p of generating the verbatim suffix of the "careless people" quote for LLAMA 1 30B ($p_{\boldsymbol{z}} \approx 35.2\%$) changes as a function of the number of independent prompts to the model with the prefix (**right**). These curves both use $1 - (1 - p_{\boldsymbol{z}})^n$ (2), plugging in the respective $p_{\boldsymbol{z}}$ (for the coin flip and for generating the "careless people" suffix given the prefix) and different values of n to compute p.

useful also for plotting distributions over $p_{\boldsymbol{z}}$ for a given book—to see how these probabilities vary for examples across a book, or for a given example across different models. We provide some examples of these comparisons in Section 4.

We also find that our version of explaining the metric is more accessible to broader audiences, which we intend to reach with this work. Hayes et al. (2025) specifically address ML and NLP audiences. We hope that our work supplements theirs in showing just how useful their metric is for capturing a probabilistic notion of extraction, and the underlying probabilistic copies that this signfies are memorized within model parameters.

A.3. Computing sequence probabilities in one forward pass

Given that the core algorithm for (n, p)-discoverable extraction is so elegantly concise, we include a (simplified) version of the code in this appendix. Our implementation differs from Hayes et al. (2025), which we confirm in discussion with the authors of that work. In that work, the authors generated the suffix a token at a time and summed up the per-generated-token conditional $\log$ probabilities. Here, we observe that this isn't necessary. We can get the logits for each token in the sequence with just one forward pass through the model, and effectively only one line of code. (See Listing 1, line 28.) For a 50-token suffix, this results in up to a $50\times$ decrease in compute—an enormous savings. (It is of course less with KV caching enabled during generation, but still nonzero). This also makes computing this metric cheaper than traditional greedy-sampled discoverable extraction (Carlini et al., 2021), which generates the greedy-sampled sequence (i.e., for a 50-token suffix, there are 50 forward passes through the model, even if those passes involve caching).

Note that this is the *entire* interaction that our main extraction experiments have with the underlying model: for a given example, we perform inference on the GPU, get the logits tensor, and move the logits to the CPU for processing. That's it. We aren't making *any* changes to the model in our extraction measurements; we are just getting the raw logits from running a given training-data example through the model. (As shown in Listing 1, we do this for a batch of examples at a time, yielding a batch of logits.)

The logits tensors we get are of shape `[batch_size, sequence_length, vocabulary_size]`. This is a three-dimensional tensor:

- Dimension 1: We have a slot for each example in the batch (with `batch_size` examples overall).
- Dimension 2: We have a position for each index in the token sequence of the example (with `sequence_length`$=100$ being the number of total positions in most of our experiments).
- Dimension 3: We have a position for each token in the whole token vocabulary for this model (with `vocabulary_size` being the overall number of positions, e.g., this is $32,000$ for the LLAMA 1 family).

So, at position `[i, j, k]` in this tensor, we have the logit (unnormalized probability) value for the `i-th` example in the batch, at the `j-th` position in the example's token sequence, for the `k-th` token in the whole token vocabulary. That is, for a given token sequence in the batch (at `[i, j]`), we have the logit values for all of the next-possible tokens in the whole token vocabulary (a `vocabulary_size`-length list of numbers, one for each token in the whole vocabulary); these values reflect the probability distribution over the next token in the sequence.

We then implement our sequence probability computations (i.e., compute $p_{\boldsymbol{z}}$) as a post-processing operation on the logits we get from the model. (See Listing 2.) That is, for any given decoding scheme, we can transform the logit distribution—for example, change the shape of the distribution according to temperature, truncate the distribution to the top-k tokens and re-normalize the distribution to have its probability still sum to 1, etc.

```
import torch
"""
Runs inference to get logits for examples z, which we will use
to compute (n,p)-discoverable extraction (i.e., to compute p_z).
  - dataloader: a torch.utils.DataLoader wrapping a torch.utils.data.Dataset containing the tokenized examples z

  - model: loaded transformers.AutoModelForCausalLM (in this snippet, assumed to have layers distributed across GPUs with device_map="auto", though for smaller models we use data parallelism)

  - returns all_logits (logits indexed by batch number). In practice, we don't do this. We asynchronously do computations on logits on CPU to save log probs.
"""
def compute_logits_for_batch(dataloader, model):

    all_logits = {}
    with torch.inference_mode():
      for batch_idx, batch in enumerate(dataloader):
        input_ids = batch["input_ids"]
        attention_mask = batch["attention_mask"]
        example_metadata_batch = batch["metadata"]

        device = model.device

        inputs = {
            "input_ids": input_ids.to(device),
            "attention_mask": attention_mask.to(device)
        }

        # Run inference; shape: [batch_size, seq_len, vocab_size]
        logits_batch = model(**inputs).logits.detach().cpu()

        input_ids = input_ids.detach().cpu()
        attention_mask = attention_mask.detach().cpu()

        all_logits[batch_idx] = logits_batch
  return all_logits
```

Listing 1: Computing logits with one forward pass through the model. This is the key speedup in our implementation and this is the only code that interacts with a given model that we study in our main extraction experiments.

This computation is written out on the right side of Equation (2). The key point here is that, for a given sequence that we are evaluating, all we do (starting at the first index in the suffix) is sum up the conditional log probability of the *actual* next token in the sequence, one at a time. That is, from the logits, we get the conditional log probabilities (Listing 2, line 19). We work with the conditional log probabilities (adding them together), rather than the conditional probabilities (and multiplying them together), as this is more numerically stable to compute. However, note that these computations are equivalent. For a given token in what we are considering as our suffix, all we are doing here is *manually* finding the conditional (log) probability associated with the *actual* next token in the sequence (Listing 2, lines 32-40). For example, consider the sequence `"the orange cat"` (where, for simplicity, we will assume each word is a single token). When processing `"the"`, the logits reflect the probabilities of the next token. We find the logit value in the distribution that corresponds to `"orange"` and add the associated conditional log probability, and then continue on (doing the same for `"orange"`, where we then get the conditional $\log$ probability for the next token being `"cat"`).

In essence, all we are doing is getting the probability of the whole sequence (by adding together the conditional $\log$ probabilities) of the actual sequence. This is $p_{\boldsymbol{z}}$: the probability of the model (under the given decoding scheme we are using) generating that sequence conditioned on the prior context (the prefix). This yields information about the relationships the model has learned from being trained on natural language—patterns in that language.

Any given sequence should be low probability. That is, for a suffix length of 50 tokens, this computation is effectively multiplying together 50 per-token conditional probabilities (by adding together conditional $\log$ probabilities). This number *should* be really small. For a sequence where each token has really high probability, this can still mean the overall sequence has low probability. Consider that each token in the suffix has conditional probability 0.9. This is *really* high; it means that a single token in the logit distribution (at each position, for the probability of the next token) has 90% of the probability mass; for a token vocabulary of size $32,000$, that means the remaining $31,999$ tokens all share the remaining 10%! For this sequence, the probability of generating it is $0.9^{50} \approx 0.005$—i.e., a 0.5% chance of generation. This is a relatively small number in the scheme of things; but it's also a really large number in general, if we consider the details of the multiplication we just did: all other possible 50-token sequences in this example are much less likely. In some cases in this paper, we observe sequences that have ***over*** 90% ***probability***—an enormous number, especially considering what the underlying computation is!

This is what memorization is: unusually high-probability sequences (where it is reasonable to consider 0.5%, or even smaller, to be unusually high). As Cooper & Grimmelmann (2024) say, "the pattern [the model has learned from the training data] *is* the memorized training data."

Also note that top-k sampling truncates the logit distribution to *only* the top-k tokens (e.g., for a vocabulary of $32,000$ tokens, there are $32,000$ logits, but top-k with $k = 40$ decoding will only consider the 40 highest-valued logits). This means that, if a given sequence has a token at a given position that is *not* in the top-k of the logit distribution, then we won't be able to complete the probability computation. This is what the `has_impossible_token` check is referring to in Listing 2, lines 36-38. In this case, the given sequence has 0 probability; it is not extractable with *any* probability.

In practice, our code deviates from Listing 2 for efficiency reasons. (The core algorithm is nevertheless the same as what we show here.) As we process conditional $\log$ probabilities, we save a lot of metadata in order to produce the plots that we include in this paper. For additional speedups (which, ultimately, total around $200\times$ prior work on (n, p)-discoverable extraction), as we compute logits on the GPU (Listing 1), we delegate our post-processing code to happen in parallel on CPU.

```python
import torch, torch.nn.functional as F, math
"""
For a batch of logits, computes the log probability (and probability)
of the suffix given the prefix for a batch of examples.
  -inputs: a batch of examples
  -logits: a batch of logits (computed doing inference, above)
  -suffix_start_idx: index in examples where suffix begins
  -transforms base distribution for temperature (temp) and top_k (k).
  -returns list of logprobs (and probs)
"""
def compute_logprob_batch(inputs, logits, suffix_start_idx, temp, k):
  input_ids = inputs_batch["input_ids"]
  attention_mask = inputs_batch["attention_mask"]
  if temp != 1.0 or k is not None:
    logits = apply_temperature_and_topk(logits, temp, k)

  # across the vocabulary, all possible next-token predictions at each
  # sequence position. shape: [batch_size, seq_len, vocab_size]
  log_probs = F.log_softmax(logits, dim=-1)
  # log_probs[b, i, j]: for example at batch index b, log probs of
  # predicting token j at position i in the sequence
  log_probs_list = [], prob_list = [], batch_size = input_ids.shape[0]
  for b in range(batch_size):
    input_ids_b = input_ids[batch_b]
    # shape: [seq_len, vocab_size]
    log_probs_b = log_probs[b]
    real_seq_len = attention_mask[b].sum().item()
    total_log_prob = 0.0, has_impossible_token = False
    # Compute log prob for each token in suffix conditioned on prefix
    for i in range(suffix_start_idx - 1, real_seq_len - 1):
      # get actual next token in suffix
      actual_next_token_id = input_ids_b[i + 1]
      # log prob for actual_next_token i+1 based on context up to i
      log_prob = log_probs_b[i, actual_next_token_id].item()
      # possible -inf from top-k filtering
      if not math.isfinite(log_prob):
        has_impossible_token = True
        break

      total_log_prob += log_prob

    if has_impossible_token:
      total_log_prob = float("-inf")
      prob = 0.0
    else:
      prob = math.exp(total_log_prob)
    log_probs_list.append(total_log_prob)
    prob_list.append(prob)

  return log_probs_list, prob_list

def apply_temperature_and_topk(logits, temp, k):
  if temperature != 1.0:
    logits = logits / temperature
  if top_k is not None:
    # top-k values from logits; shape: [batch, seq_len, top_k]
    top_k_vals = torch.topk(logits_batch, top_k, dim=-1)
    # k-th element in top-k values for each item in batch
    kth_vals = top_k_vals.values[:, :, -1].unsqueeze(-1)
    mask = logits < kth_vals
    logits = logits.masked_fill(mask, float("-inf"))
  return logits
```

Listing 2: Computing logprobs

B. Additional notes on memorization, extraction risk, and U.S. copyright

Not all examples are extractable. We can see this by looking at the maximum probabilistic extraction rate. (See Appendix E.) Because there is a logarithmic relationship between n and the p—that is, $n \propto \log(1-p)$—there is a maximum total extraction rate that we can get. One can see this by looking Equation (2):

Proof. Let $p_{\boldsymbol{z}} \in [0,1]$ denote the probability of extracting example $\boldsymbol{z}$ in a single trial, i.e., one prompt. Then the probability of not extracting $\boldsymbol{z}$ in a single trial is $1 - p_{\boldsymbol{z}}$, and the probability of never extracting $\boldsymbol{z}$ after n independent trials is:

$$(1 - p_{\boldsymbol{z}})^n$$

Therefore, the probability of extracting $\boldsymbol{z}$ at least once in n trials is:

$$\mathbb{P}^{\boldsymbol{z}}_{\text{extracted}}(n) = 1 - (1 - p_{\boldsymbol{z}})^n$$

Now consider two cases:

- If $p_{\boldsymbol{z}} > 0$, then $\lim_{n\to\infty}(1-p_{\boldsymbol{z}})^n = 0$, and thus $\lim_{n\to\infty} \mathbb{P}^{\boldsymbol{z}}_{\text{extracted}}(n) = 1$. (Of course, if $p_{\boldsymbol{z}} = 1$, then $\mathbb{P}^{\boldsymbol{z}}_{\text{extracted}}(n) = 1$ for all n.)
- If $p_{\boldsymbol{z}} = 0$, then $\mathbb{P}^{\boldsymbol{z}}_{\text{extracted}}(n) = 0$ for all n.

Now suppose we have a finite set of examples $\mathcal{Z}$, each with its own extraction probability $p_{\boldsymbol{z}} \in [0,1]$. (This is how extraction rates are computed in practice, for such a finite set of examples $\mathcal{Z}$.) The expected number of *distinct* examples extracted at least once over $\boldsymbol{z}$ trials is:

$$\mathbb{E}[\text{\# unique extractions}] = \sum_{\boldsymbol{z}\in\mathcal{Z}} \left[1 - (1 - p_{\boldsymbol{z}})^n\right]$$

Taking the limit as $n \to \infty$, we get:

$$\lim_{n\to\infty} \mathbb{E}[\text{\# unique extractions}] = \sum_{\boldsymbol{z}\in\mathcal{Z}} \mathbb{I}[p_{\boldsymbol{z}} > 0]$$

where $\mathbb{I}[p_{\boldsymbol{z}} > 0]$ is the indicator function, equal to 1 when $p_{\boldsymbol{z}} > 0$, and 0 otherwise.

Therefore, when some examples may be unextractable (i.e., $p_{\boldsymbol{z}} = 0$), the maximum number of unique extractions is given by:

$$|\mathcal{Z}_{\text{extractable}}| := |\{\boldsymbol{z} \in \mathcal{Z} \mid p_{\boldsymbol{z}} > 0\}|$$

And so, even with an infinite number of trials, the number of distinct examples ever extractable is bounded by $|\mathcal{Z}_{\text{extractable}}| \leq |\mathcal{Z}|$. This means, for this distinct set, there is a maximum probabilistic extraction rate, which is independent of the choice of p, in (n,p)-discoverable extraction. □

We can see this in practice for plots that examine the maximum extraction rate, e.g., see Appendix C.

Unextractable sequences in practice. There are various ways that a given example $\boldsymbol{z}$ could have $p_{\boldsymbol{z}} = 0$, thereby limiting the total extraction rate. For one, if we use top-k sampling (as we do in this paper, with $k = 40$), not all tokens in the vocabulary are reachable during sample (e.g., for LLAMA models, only 40 out of $32,000$ tokens would be able to be generated in a given sampling iteration). When computing (n,p)-discoverable extraction (see 2), if the actual token in the target suffix that we are summing over (summing their conditional $\log$ probabilities) isn't in the top 40 tokens, then it becomes impossible to generate the target suffix. In other words, that example is not extractable (with respect to this model and decoding scheme).

For another, numbers are represented in computers with a finite number of bits. This means that there is a minimum number that can be represented (> 0) before it gets rounded down to 0. For instance, imagine a token vocabulary of $32,000$ tokens (as with LLAMA models). If the model just generated tokens completing randomly (a true monkey at the typewriter (The Simpsons: Last Exit to Springfield (Fox television broadcast Mar. 11, 1993); Borel, 1913)), then at each iteration every token would have probability $\frac{1}{32000}$. When generating 71 such tokens in a row (i.e., a 71-token sequence), the probability of any such sequence is ${\frac{1}{32000}}^{71} \approx 1.3626 \times 10^{-320}$. Once we try to generate a 72nd token, the probability for the whole sequence becomes ${\frac{1}{32000}}^{72} = 0$, in terms of how the probability gets represented in the computer. There is underflow (i.e., rounding down to 0). As a result, even with temperature sampling alone, not every sequence (for long enough sequences) is extractable in practice.

Extractability and copyright. Even though not every sequence is extractable, this doesn't mean that sequences that have very low extraction probabilities are memorized in a sense that is of use to copyright. The important point to distinguish is if the sequence $\boldsymbol{z}$ in question is more likely to be generated than is expected from the base distribution of the model (and the given decoding scheme). While we don't attempt to draw such a line in this paper, we note that Hayes et al. (2025) does experiments to validate their (n, p)-discoverable extraction metric. They compare the extraction of training data to the rate of, by chance, generating test data (unseen data, that cannot by definition have been memorized). Put differently, they empirically test the "monkey at a typewriter" problem, to see if this is having an effect on their measurements—to confirm that their measurements of extraction are catching true instances of memorization.

They find that it takes orders of magnitude more queries to generate test data (at very large n) than it does to reliably extract training data, indicating that the metric is capturing valid instances of memorization. Of course, this also suggests that there exists a large enough n at which it becomes challenging to distinguish truly extracted training data versus spontaneous generation of test data. In this paper, we avoid being in this regime by choosing a very conservative minimum value of $p_{\boldsymbol{z}}$ (and thus, a capped maximum expected n) in our plots. Every number we report in the main paper is for $p_{\boldsymbol{z}} \leq 0.0001 = 0.01\%$. We show a few instances of extractions that have smaller $p_{\boldsymbol{z}}$ than this in Appendix G, for illustrative purposes.

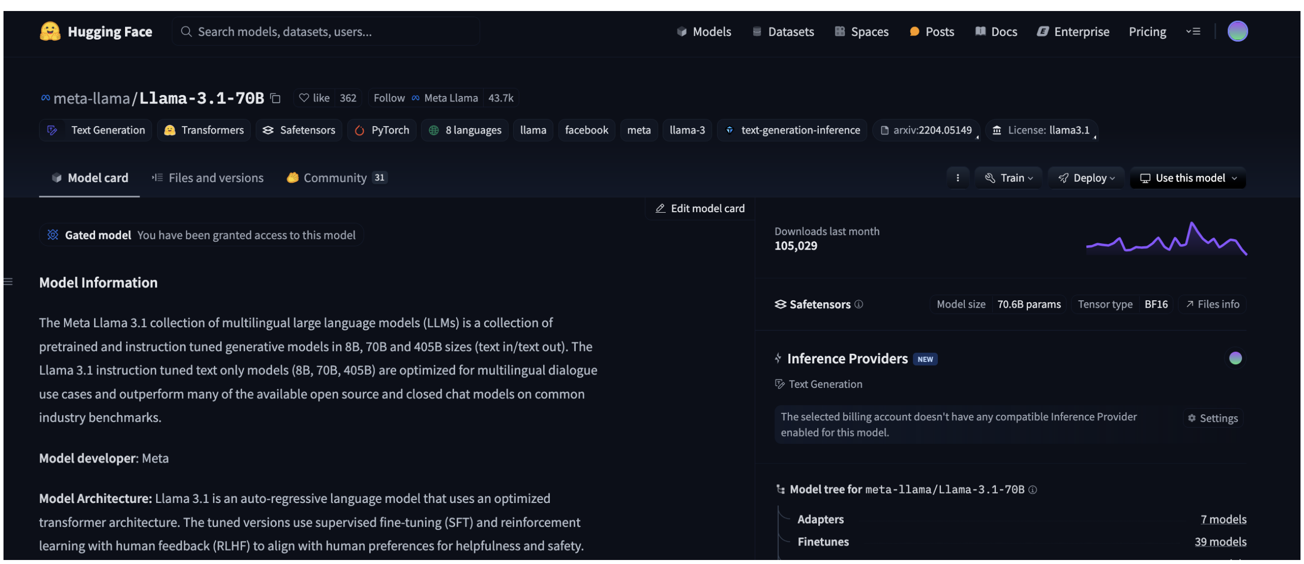


Figure 10: Screenshot of the month's downloads of LLAMA 3.1 70B on HuggingFace, taken by the authors in May 2025.

C. Testing our measurement pipeline

To confirm the accuracy of our implementation of (n, p)-discoverable extraction, as well as to benchmark its efficiency in comparison to the implementation in Hayes et al. (2025), we re-run the experiments from Hayes et al. (2025) for small PYTHIA models on the Enron dataset. We discussed the original run times and sample of Enron with the authors of Hayes et al. (2025) to ensure consistency. We provide some brief details about these experiments to show due diligence for our implementation in our copyright-relevant setting.

We ran these experiments with both `float16` and `float32`, and report results for `float32` to align with Hayes et al. (2025). (However, we note no large differences, an observation we defer to separate work). In Table 1, we show the runtime for our experiments on the $10,000$, 100-token examples Hayes et al. (2025) drew from Enron. In Table 2, we report the same metrics as Hayes et al. (2025): the greedy-sampled discoverable extraction rate and (as a point of comparison) the maximum (n, p)-discoverable extraction rate. We also include extraction curves in Figure 11, which show the same settings as the same experiments in Hayes et al. (2025). We run these experiments on the same 4 A100s as all of our other experiments.

	Time (mm:ss)	**Parallelism**	**Batch size**
PYTHIA 1B	02:30.42	Data (4 GPUs)	250
PYTHIA 2.8B	03:36.35	Data (4 GPUs)	250

Table 1: Runtime for different PYTHIA models on a $10,000$-example dataset drawn from Enron.

PYTHIA 1B on Enron subset	**Hayes et al. (2025) (No BOS token)**	**With BOS token**
Greedy extraction rate	0.76%	0.74%
Max. (n, p)-discoverable extraction rate	5.27%	5.52%
PYTHIA 2.8B on Enron subset	**Hayes et al. (2025) (No BOS token)**	**With BOS token**
Greedy extraction rate	1.3 %	1.82%
Max. (n, p)-discoverable extraction rate	9.04%	9.47%

Table 2: Greedy discoverable extraction and maximum (n, p)-discoverable extraction rates for PYTHIA models on a $10,000$-example subset from Enron. We replicate Hayes et al. (2025), which did not include the beginning of sequence (`BOS`) token in the examples they extract for their experiments using `Pythia` models. We compare to the extraction rates where the `BOS` token is included at the start of the example. Note that the (n, p)-discoverable extraction rate increases with the presence of the `BOS` token. See also Figure 11.

In the process of running these experiments, we identified a small bug in Hayes et al. (2025). The PYTHIA tokenizer does *not* add a beginning of sequence (`BOS`) token by default to the sequences it tokenizes. This is reflective of GPT-2-style tokenizers. However, for our setting, it's best practice to include this token at the start, as we';; be submitting these sequences as input for inference to LLMs (that are trained to expect this token at the beginning of the sequence). When we manually prepend this token and re-run the experiments from Hayes et al. (2025), we observe elevated extraction rates compared to Hayes et al. (2025). See Table 2. We similarly will prepend the `BOS` token in our new experiments using PYTHIA models.

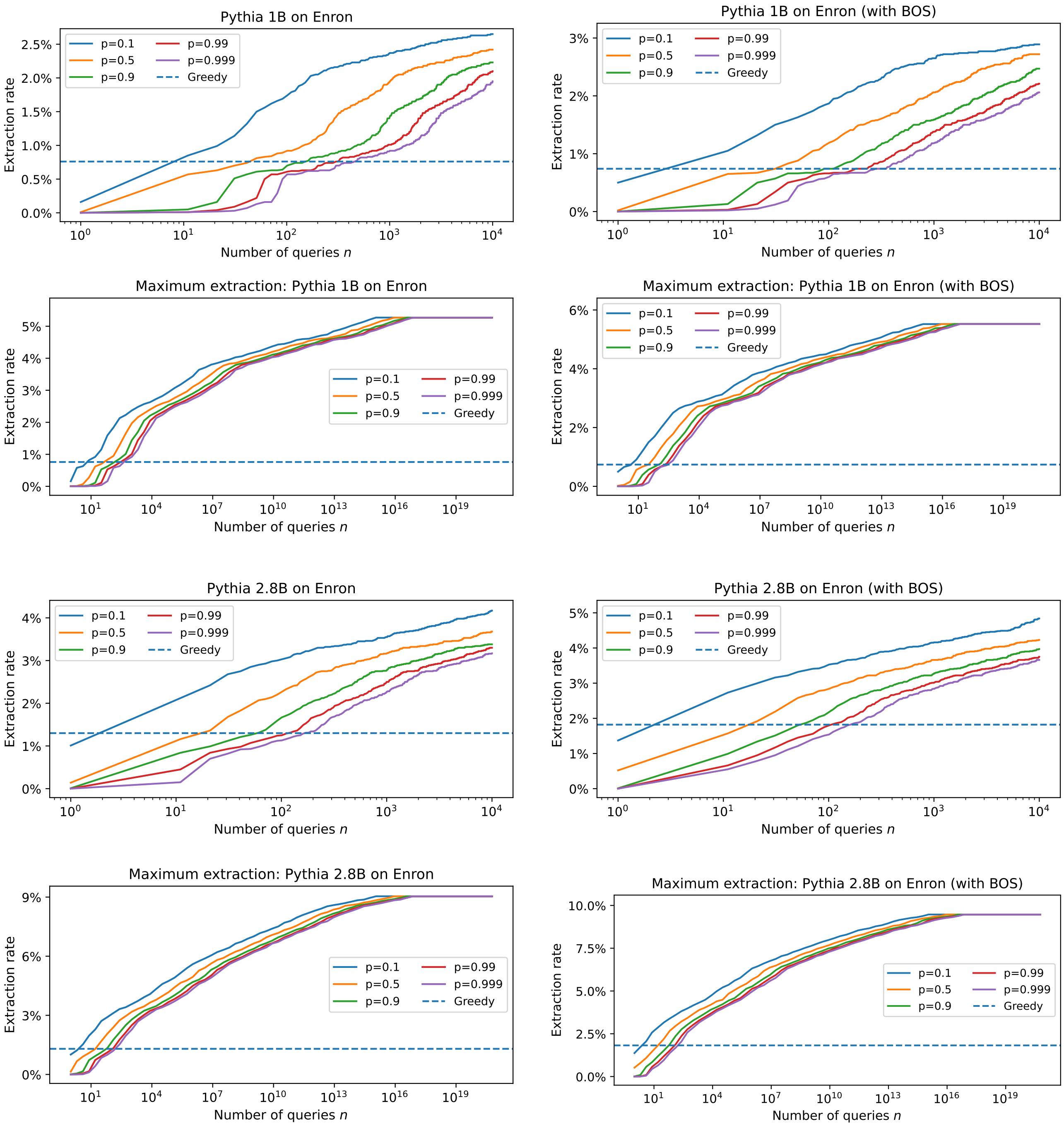


Figure 11: Replicating Hayes et al. (2025): Extraction curves for PYTHIA 1B and PYTHIA 2.8B on a $10,000$-example sample of Enron, where each example is 100 tokens (50/50 prefix/suffix split). These results are for temperature $T = 1$, top-k with $k = 40$ sampling. We show to sets of plots for each model. The left column is identical to Hayes et al. (2025); the right column shows the same set of curves, but for the version of the experiment where we manually prepend the BOS token to each example.

D. Details on experimental settings

In this Appendix, we discuss overall experimental setup details. We describe additional experimental details in different appendies about those experiments.

Data. In all of our experiments for this project, we draw our prompts (prefixes) from `Books3`; we verify extraction against the corresponding suffixes drawn from `Books3`. We obtained this dataset from a previous (2022) download of The Pile (Gao et al., 2020), which is stored on a university cluster and which we use for research purposes on language modeling.

The status of `Books3` as a research artifact (as well as a training corpus for language models) remains unresolved. (See Section 3.) We note that this dataset remains widely available in the research community; for example, found in the HuggingFace-hosted version of The Pile (EleutherAI/the_pile_deduplicated), as well as in various other data repositories (e.g., SaylorTwift/the_pile_books3_minus_gutenberg; amongglue/books3-subset-raw; CANBERT/pile_books3_text). It also features as part of a benchmark task in the popular HELM evaluation suite.

We sample this dataset at random (with caveats) for our overall extraction rate experiments in Section 3. We detail these experiments in Appendix E. We work with a specific set of books for our book-specific experiments. In Table 3, we list the books in our sliding-window experiments. (See Section 4 and Appendix F.1.) In a second set of experiments, we attempted (and confirmed) that we would obtain reasonable approximations of per-book extraction rates with a cheaper sampling strategy. The sliding-window approach remains the most fruitful for completeness of identifying memorization hot-spots within books. We therefore omit these other results.

Book selection. We began our project with *The Great Gatsby* (Fitzgerald, 1925). This choice stems from a conversation in 2023 while writing a different paper, when Anthropic announced its 100K-length context for Claude with the example of prompting with the entirety of *The Great Gatsby* (Anthropic, 2023).

We then continued our selection process with the books listed with the associated plaintiffs in the (amended) class action complaint of Kadrey et al. v. Meta (Kadrey et al. v. Meta Platforms, Inc., pp. 4-5). This is how we sourced the first 13 books from `Books3`. We expanded to additional books by these plaintiffs, and also added in some generally popular books that we chose among the team (e.g., Rowling (1998); Tolkien (1937); Camus (1955); Heller (1961)) and some (less publicly popular) academic books based on our personal preferences (e.g., Zittrain (2008); Barolini (2006)). (This team, after all, is composed of academics.) We deliberately include selections from the public domain (e.g., Carroll (1865); Shelley (1818); Woolf (1928)), and an example of a book that was published under a permissive CC license (Doctorow (2003)). We added in Carroll (1865) because of concurrent work that was published on arXiv just as we were about to post our own: Ma et al. (2025). To round out the list, we also selected books at random from the manifest file that accompanies the `Books3` archive (e.g., Jacobs (1999), Rouighi (2011), Cogburn (2012)). Once we observed that certain mainstream-popular books were highly memorized by LLAMA 3.1 70B, we added in 5 additional such books to test: Martin (1996), Brown (2003), Meyer (2005), Orwell (1949), and Sandberg (2013).

Models. Altogether, we run experiments on 17 continuation-style (i.e., non-chatbot) models. Our sliding window experiments (Appendix F.1) on specific books use 10 models: PYTHIA 12B (Biderman et al., 2023b), PHI 4 (a 14B model) (Abdin et al., 2024), LLAMA 1 13B, LLAMA 1 65B (Touvron et al., 2023a), LLAMA 2 70B (Touvron et al., 2023b), LLAMA 3 70B, LLAMA 3.1 70B (Grattafiori et al., 2024), DEEPSEEK V1 67B (DeepSeek-AI et al., 2024), QWEN 2.5 72B (Qwen et al., 2025), and GEMMA 2 27B (Team et al., 2024). We our average extraction experiments on a subset of these models (Appendix E), as well as 6 smaller models (LLAMA 1 7B, LLAMA 2 7B, LLAMA 3.1 8B, PYTHIA 6.9B, DEEPSEEK V1 7B, and GEMMA 2 9B) and 1 additional 13B model (LLAMA 2 13B). We also ran some initial sliding window tests on these 7 models, but ultimately opted to exclude them from our analysis. The smaller models don't memorize very much content from specific books (at least, not the books we tested). We predominantly focus on larger models in those experiments.

We use `float16` for all models except for LLAMA 3, LLAMA 3.1, and QWEN 2.5 models. These models were explicitly trained to work with `bfloat16`. In an earlier version of this work, we reported results for LLAMA 3.1 that used `float16`. Slight differences in results in this version are attributed to the change to `bfloat16`. The overall conclusions remain the same.

Compute resources.. We run all of our experiments in a slurm cluster environment, using the same node with 4 A100 GPUs.

	Author	Title	Year	Status	`Books3` path
1	Margaret Atwood	*The Handmaid's Tale* (Atwood, 1985)	1985	©	the-eye.eu/public/Books/Bibliotik/M/Margaret Atwood (1985) The Handmaid˙s Tale [retail].epub.txt
2	Teodolinda Barolini	*Dante and the Origins of Italian Literary Culture* (Barolini, 2006)	2006	©	the-eye.eu/public/Books/Bibliotik/D/Dante and the Origins of Italia - Barolini, Teodolinda;.epub.txt
3	Dan Brown	*The Da Vinci Code* (Brown, 2003)	2003	©	the-eye.eu/public/Books/Bibliotik/D/Dan Brown - The Da Vinci Code.epub.txt
4	Albert Camus (Justin O'Brien, translator)	*The Myth of Sisyphus* (Camus, 1955)	1955	©	the-eye.eu/public/Books/Bibliotik/2/2013(orig1942) Albert Camus - The Myth of Sisyphus[Transl Justin O'Brien]˙Ral.epub.txt
5	Lewis Carroll	*Alice's Adventures in Wonderland* (Carroll, 1865)	1865	PD	the-eye.eu/public/Books/Bibliotik/A/Alice's Adventures in Wonderlan - Lewis Carroll.epub.txt
6	Ta-Nehisi Coates	*The Beautiful Struggle* (Coates, 2009)	2009	©	the-eye.eu/public/Books/Bibliotik/T/The Beautiful Struggle - Ta-Nehisi Coates.epub.txt
7	Ta-Nehisi Coates	*We Were Eight Years in Power: An American Tragedy* (Coates, 2017)	2017	©	the-eye.eu/public/Books/Bibliotik/W/We Were Eight Years in Power - Ta-Nehisi Coates.epub.txt
8	Ta-Nehisi Coates	*The Water Dancer: A Novel* (Coates, 2019)	2019	©	the-eye.eu/public/Books/Bibliotik/T/The Water Dancer - Ta-Nehisi Coates;.epub.txt
9	Jon Cogburn	*Dungeons and Dragons and Philosophy* (Cogburn, 2012)	2012	©	the-eye.eu/public/Books/Bibliotik/2/2012 Jon Cogburn - Dungeons and Dragons and Philosophy - Raiding the Temple of Wisdom˙Rsnl.epub.txt
10	Junot Díaz	*Drown* (Díaz, 1996)	1996	©	the-eye.eu/public/Books/Bibliotik/J/Junot Diaz - Drown.epub.txt
11	Junot Díaz	*The Brief Wondrous Life of Oscar Wao* (Díaz, 2007)	2007	©	the-eye.eu/public/Books/Bibliotik/T/The Brief Wondrous Life of Oscar Wao.epub.txt
12	Junot Díaz	*This Is How You Lose Her* (Díaz, 2012)	2012	©	the-eye.eu/public/Books/Bibliotik/T/This Is How You Lose Her-Diaz.epub.txt
13	Cory Doctorow	*Down and Out in the Magic Kingdom* (Doctorow, 2003)	2003	CC-BY-NC-SA	the-eye.eu/public/Books/Bibliotik/D/Down and Out in the Magic Kingd - Cory Doctorow.epub.txt
14	Carol Ann Duffy	*The World's Wife* (Duffy, 2001)	2001	©	the-eye.eu/public/Books/Bibliotik/T/The World˙s Wife - Carol Ann Duffy.epub.txt
15	Jennifer Egan	*A Visit from the Goon Squad* (Egan, 2010)	2010	©	the-eye.eu/public/Books/Bibliotik/A/A“ Visit“ from“ the“ Goon“ Squad“ -“ Jennifer“ Egan.epub.txt
16	Christopher Farnsworth	*The President's Vampire* (Farnsworth, 2011)	2011	©	the-eye.eu/public/Books/Bibliotik/T/The President's Vampire - Christopher Farnsworth.epub.txt
17	F. Scott Fitzgerald	*The Great Gatsby* (Fitzgerald, 1925)	1925	PD	the-eye.eu/public/Books/Bibliotik/T/The Great Gatsby - F. Scott Fitzgerald.epub.txt
18	Malcom Gladwell	*Blink: The Power of Thinking Without Thinking* (Gladwell, 2005)	2005	©	the-eye.eu/public/Books/Bibliotik/B/Blink - Malcolm Gladwell.epub.txt
19	Christopher Golden	*Dead Ringer* (Golden, 2016)	2016	©	the-eye.eu/public/Books/Bibliotik/C/Christopher Golden - Dead Ringers [retail].epub.txt
20	Christopher Golden	*Ararat* (Golden, 2017)	2017	©	the-eye.eu/public/Books/Bibliotik/A/Ararat - Christopher Golden.epub.txt
21	Andrew Sean Greer	*The Confessions of Max Tivoli*	2005	©	the-eye.eu/public/Books/Bibliotik/T/The Confessions of Max Tivoli - Andrew Sean Greer.epub.txt
22	John Grisham	*Theodore Boone: The Fugitive* (Grisham, 2015)	2015	©	the-eye.eu/public/Books/Bibliotik/T/The Fugitive - John Grisham.epub.txt
23	Mark Haddon	*The Curious Incident of the Dog in the Night-Time* (Haddon, 2003)	2003	©	the-eye.eu/public/Books/Bibliotik/T/The Curious Incident of the Dog in the Night-Time - Mark Haddon.epub(1).txt
24	Joseph Heller	*Catch-22* (Heller, 1961)	2018	©	the-eye.eu/public/Books/Bibliotik/C/Catch-22 - Joseph Heller.epub.txt
25	David Henry Hwang	*M. Butterfly* (Hwang, 1988)	1988	©	the-eye.eu/public/Books/Bibliotik/M/M. Butterfly - David Henry Hwang.epub.txt
26	Betty E.M. Jacobs	*All the Onions* (Jacobs, 1999)	1999	©	the-eye.eu/public/Books/Bibliotik/A/All the Onions - Betty E. M. Jacobs.epub.txt
27	James Joyce	*Ulysses* (Joyce, 1922)	1922	PD	the-eye.eu/public/Books/Bibliotik/U/Ulysses - James Joyce - Penguin Group - 2000.epub.txt
28	Richard Kadrey	*Sandman Slim* (Kadrey, 2009)	2009	©	the-eye.eu/public/Books/Bibliotik/S/Sandman Slim - Richard Kadrey.epub.txt
29	Matthew Klam	*Who Is Rich?* (Klam, 2017)	2017	©	the-eye.eu/public/Books/Bibliotik/W/Who Is Rich˙ - Matthew Klam.epub.txt
30	Laura Lippman	*After I'm Gone* (Lippman, 2014)	2014	©	the-eye.eu/public/Books/Bibliotik/A/After I'm Gone - Laura Lippman.epub.txt
31	Laura Lippman	*Sunburn* (Lippman, 2018)	2018	©	the-eye.eu/public/Books/Bibliotik/S/Sunburn - Laura Lippman.epub.txt
32	George R.R. Martin	*A Game of Thrones* (Martin, 1996)	1996	©	the-eye.eu/public/Books/Bibliotik/A/A Game of Thrones - George R. R. Martin.epub.txt
33	Stephenie Meyer	*Twilight* (Meyer, 2005)	2005	©	the-eye.eu/public/Books/Bibliotik/T/Twilight - Stephenie Meyer.epub.txt
34	Toni Morrison	*Beloved* (Morrison, 1987)	1987	©	the-eye.eu/public/Books/Bibliotik/B/Beloved - Toni Morrison.epub.txt
35	Yōko Ogawa	*The Memory Police* (Ogawa, 2019)	2019	©	the-eye.eu/public/Books/Bibliotik/T/The Memory Police - Yoko Ogawa.epub.txt
36	George Orwell	*Nineteen-Eighty Four* (Orwell, 1949)	1949	©	the-eye.eu/public/Books/Bibliotik/N/Nineteen Eighty-Four (The Annotated Edition) - George Orwell.epub.txt
37	Philip Pullman	*The Subtle Knife* (Pullman, 1997)	1997	©	the-eye.eu/public/Books/Bibliotik/P/Pullman - The Subtle Knife.epub.txt
38	Ramzi Rouighi	*The Making of a Mediterranean Emirate* (Rouighi, 2011)	2011	©	the-eye.eu/public/Books/Bibliotik/T/The Making of a Mediterranean E - Rouighi, Ramzi;.epub.txt
39	J.K. Rowling	*Harry Potter and the Sorcerer's Stone* (Rowling, 1998)	1998	©	the-eye.eu/public/Books/Bibliotik/H/Harry˙Potter˙and˙the˙Sorcerers˙Stone-Rowling.epub.txt
40	J.K. Rowling	*Harry Potter and the Goblet of Fire* (Rowling, 2000)	2000	©	the-eye.eu/public/Books/Bibliotik/H/Harry˙Potter˙and˙the˙Goblet˙of˙Fire-Rowling.epub.txt
41	J.D. Salinger	*The Catcher in the Rye* (Salinger, 1951)	1951	©	the-eye.eu/public/Books/Bibliotik/T/The Catcher in the Rye - J. D. Salinger.epub.txt
42	Sheryl Sandberg	*Lean In* (Sandberg, 2013)	2013	©	the-eye.eu/public/Books/Bibliotik/L/Lean In - Sheryl Sandberg.epub.txt
43	Sarah Silverman	*The Bedwetter* (Silverman, 2010)	2010	©	the-eye.eu/public/Books/Bibliotik/T/The Bedwetter - Sarah Silverman.epub.txt
44	Rachel Louise Snyder	*No Visible Bruises* (Snyder, 2019)	2019	©	the-eye.eu/public/Books/Bibliotik/N/No Visible Bruises - Rachel Louise Snyder.epub.txt
45	Lysa TerKeurst	*Unglued: Making Wise Choices in the Midst of Raw Emotions* (TerKeurst, 2012)	2012	©	the-eye.eu/public/Books/Bibliotik/L/Lysa TerKeurst, Unglued.epub.txt
46	Lysa TerKeurst	*Embraced* (TerKeurst, 2018)	2018	©	the-eye.eu/public/Books/Bibliotik/E/Embraced - Lysa Terkeurst.epub.txt
47	J.R.R. Tolkien	*The Hobbit* (Tolkien, 1937)	1937	©	the-eye.eu/public/Books/Bibliotik/T/The Hobbit (Houghton Mifflin Harcourt) (75th Anniversary Edition) [Epub] - J.R.R. Tolkien.epub.txt
48	Jacqueline Woodson	*Brown Girl Dreaming* (Woodson, 2014)	2014	©	the-eye.eu/public/Books/Bibliotik/B/Brown Girl Dreaming - Jacqueline Woodson.epub.txt
49	Jacqueline Woodson	*Another Brooklyn* (Woodson, 2016)	2016	©	the-eye.eu/public/Books/Bibliotik/A/Another Brooklyn - Jacqueline Woodson.epub.txt
50	Jonathan Zittrain	*The Future of the Internet and How to Stop It* (Zittrain, 2008)	2008	©	the-eye.eu/public/Books/Bibliotik/J/Jonathan Zittrain - The Future of the Internet.epub.txt

Table 3: The 50 Books (drawn from `Books3`) that we sampled for our book-specific sliding window extraction experiments. See Section 4 and Appendix F.1.

E. Additional information on extraction rates

For our experiments that produce average extraction rates over random samples of `Books3`, we do 3 separate runs on $40,000$ examples on 13 different models: LLAMA 1 7B, LLAMA 1 13B, LLAMA 1 65B (Touvron et al., 2023a), LLAMA 2 7B, LLAMA 2 13B, LLAMA 2 70B (Touvron et al., 2023b), LLAMA 3.1 8B, LLAMA 3.1 70B (Grattafiori et al., 2024), PYTHIA 6.9B, PYTHIA 12B (Biderman et al., 2023b), DEEPSEEK V1 7B (DeepSeek-AI et al., 2024), GEMMA 2 9B (Team et al., 2024), and PHI 4 (Abdin et al., 2024) (which is a 14B model). We group these models (roughly) into three sizes: small, medium and large.

Sampling procedure. For each run, we sample $4,000$ books without replacement from `Books3`. Then, from each of these books, we sample 10 non-overlapping 100-token examples, for a total of $40,000$ examples. We make sure that the start token is a space, to ensure that we are not starting in an "odd" spot—one that could lead to unusual tokenization. We run these $40,000$ examples through each of the 13 models to compute (n, p)-discoverable extraction metrics. We follow this procedure 3 separate times (i.e., for a total of $120,000$ examples across $12,000$ different books) to get a sense of the variance across extraction metrics for different samples. We report results for an example size of 100 tokens (50/50 prefix/suffix split) for greedy-sampled discoverable extraction and (n, p)-discoverable extraction with temperature $T = 1$ and top-k sampling with $k = 40$.

We summarize the results in Table 4. This table shows the maximum possible (n, p)-discoverable extraction rate, compared to the greedy-sampled discoverable extraction rate for each model. We show the average and ± 1 standard deviation over the 3 different trials. We also plot these average rates (and a third point—the (n, p)-discoverable extraction rate for $p \geq 0.01\%$) in Figure 12. We show more detailed extraction-rate curves (as a function of n and for various settings of p) in Figure 13. Each run of $40,000$ examples took between approximately 8 minutes and 45 minutes on 4 A100s. We omit detailed timing results for brevity.

High-level takeaways.. As we expected (and discuss in Section 3), *average* extraction rates are relatively low. The minimum (n, p)-discoverable extraction rate that we observe (for $p \geq 0.01\%$) is for PHI 4 (under 0.2%); the largest is for LLAMA 3.1 70B (around 0.75%).

Aligning with prior work, larger models tend to memorize more than smaller ones (Hayes et al., 2025; Carlini et al., 2023b; Nasr et al., 2023; 2025). For our results, this is true within the same model family, and also across model families. PHI 4 is the exception; it is a 14B model with an extraction rate that is on par with the small (7-9B) models that we test. Unlike the other models we test, PHI 4 was trained predominantly on synthetic data; based on the PHI 4 technical report (Abdin et al., 2024), `Books3` should not have been included in its training data. We are still able to extract data that is in `Books3`, and we investigate this further in Appendix G.

Later generations of LLAMA model families tend to memorize more `Books3`-contained text. For just one example, LLAMA 3.1 70B memorizes more on average than LLAMA 2 70B, which memorizes more on average than LLAMA 1 65B. This pattern doesn't always hold for specific books. We defer additional, related observations about LLAMA 3.1 70B to Appendix H.

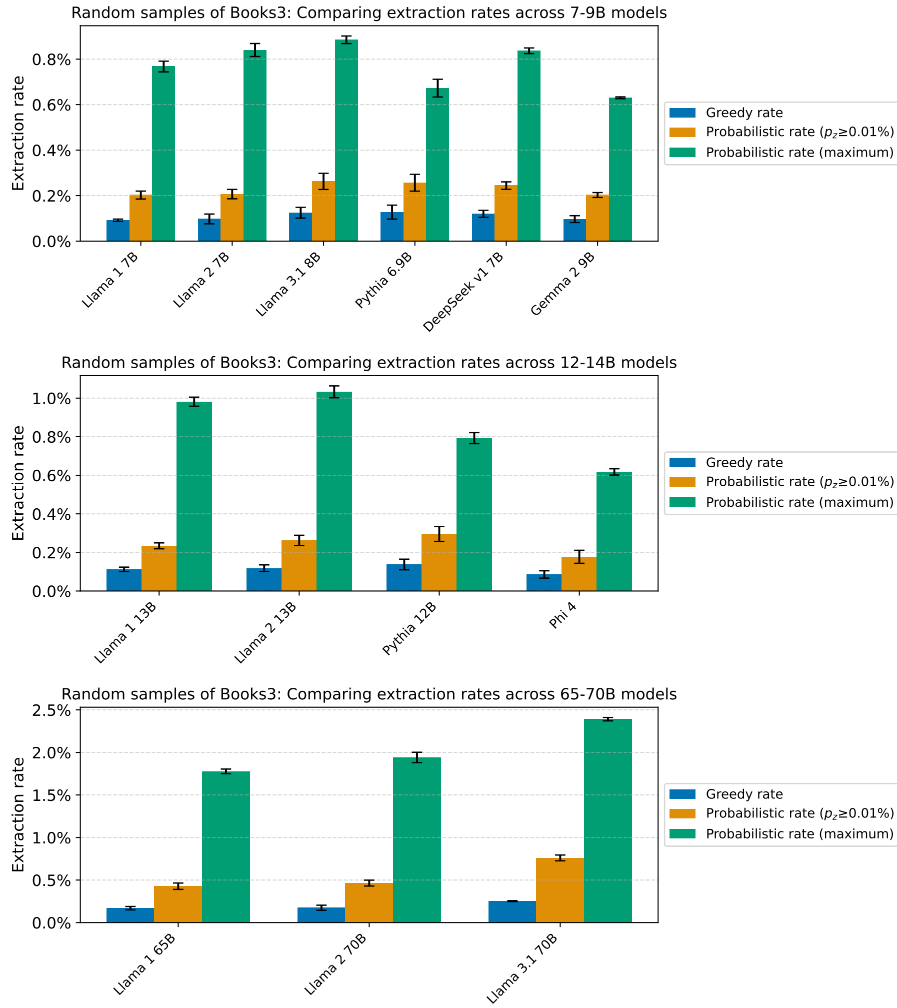


Figure 12: Comparing extraction rates for small (7-9B), medium (12-14B), and large models (65-70B) As expected, *average* extraction rates on `Books3` are relatively low for all models. Aligning with prior work, we observe that larger models memorize more. For our setting, this is true for both within and across model families. PHI 4, a 14B model, is an exception: its extraction rates are on par with smaller models. More recent generations of LLAMA family models memorize more than previous ones. Error bars here indicate variance across the 3 samples of $40,000$ examples that we run.

Model	**Small (7–9B)**		**Medium (12–14B)**		**Large (65–70B)**	
	Greedy rate	**Max (n, p) rate**	**Greedy rate**	**Max (n, p) rate**	**Greedy rate**	**Max (n, p) rate**
LLAMA 1	$0.09 \pm 0.01\%$	$0.77 \pm 0.02\%$	$0.11 \pm 0.01\%$	$0.98 \pm 0.02\%$	$0.17 \pm 0.02\%$	$1.78 \pm 0.03\%$
LLAMA 2	$0.10 \pm 0.02\%$	$0.84 \pm 0.03\%$	$0.12 \pm 0.02\%$	$1.03 \pm 0.03\%$	$0.17 \pm 0.03\%$	$1.94 \pm 0.06\%$
LLAMA 3.1	$0.12 \pm 0.02\%$	$0.89 \pm 0.02\%$	–	–	$0.25 \pm 0.01\%$	$2.39 \pm 0.02\%$
PYTHIA	$0.13 \pm 0.03\%$	$0.67 \pm 0.04\%$	$0.14 \pm 0.03\%$	$0.79 \pm 0.03\%$	–	–
DEEPSEEK V1	$0.12 \pm 0.02\%$	$0.84 \pm 0.01\%$	–	–	–	–
GEMMA 2	$0.10 \pm 0.02\%$	$0.63 \pm 0.00\%$	–	–	–	–
PHI 4	–	–	$0.09 \pm 0.02\%$	$0.62 \pm 0.02\%$	–	–

Table 4: Greedy and maximum probabilistic extraction rates (in %) per model family and size. See Figures 12 & 13 for more detailed extraction-rate curves.

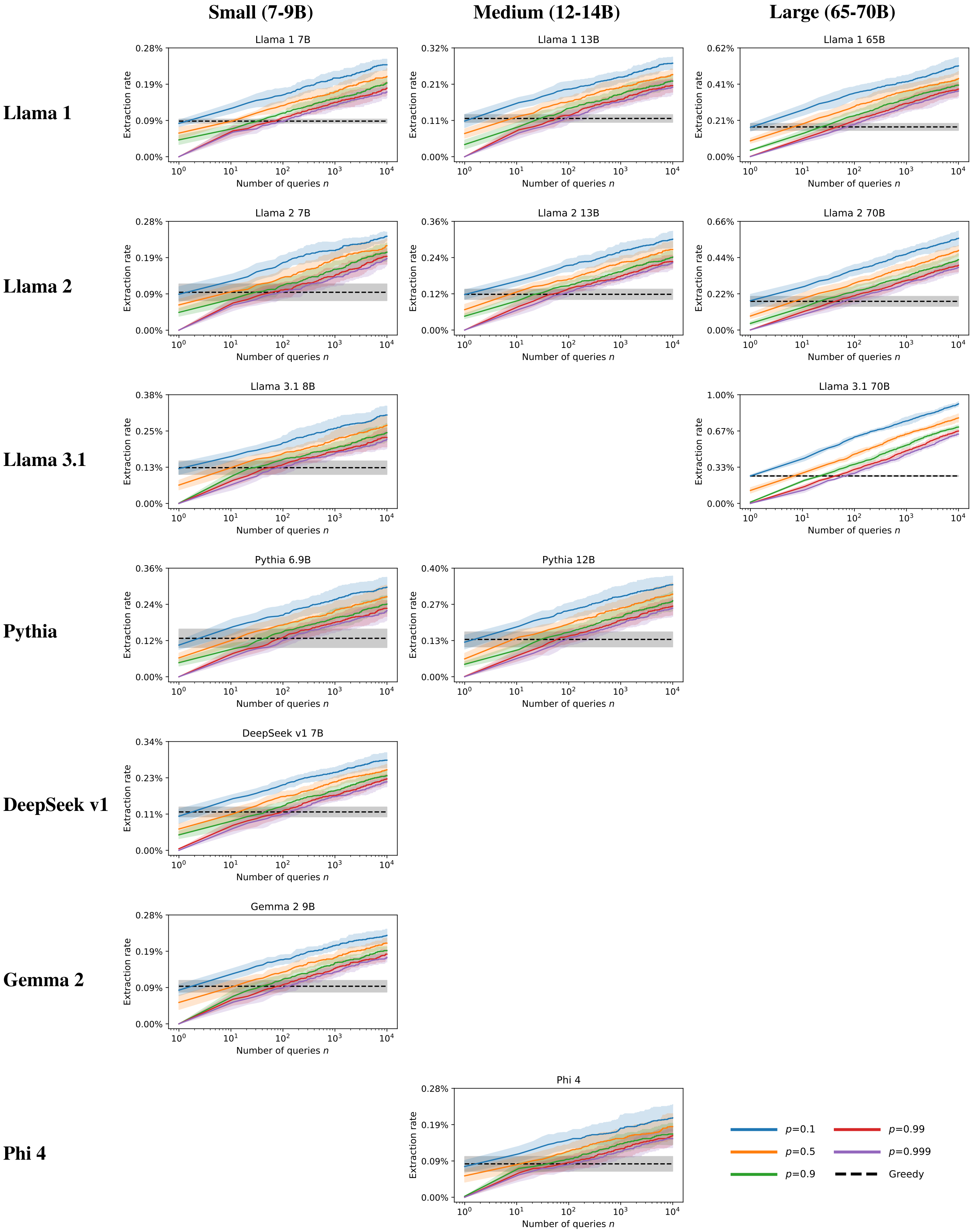


Figure 13: Extraction rates for different models averaged over 3 different samples of 100-token examples drawn from `Books3`. See Appendix E main text, Table 4, and Figure 12 for more details.

F. Additional results for per-book extraction experiments

In this appendix, we provide extensive details on our "panning for gold" extraction experiments. That is, these experiments take the approach of building examples $\boldsymbol{z}$ out of `Books3` books (to attempt to extract) using a sliding-window approach (Appendix F.1). We first discuss the overall setup, and then provide per-book results.

F.1. Setup: Sliding-window experiments

For each of the 50 books for which we run sliding-window experiments. (See Table 3 for the full list in one summarized format.) These experiments are meant to help us, effectively, "pan for gold": to identify regions within specific books where there are high-probability stretches of memorized content. The approach that we take reveals the position (in character space) in each book where these regions occur (and if they occur at all).

We discuss the sampling configurations we use for each of these experiments, then detail the sampling procedure. In our discussion of the sampling procedure, we take care to explain what the intention is behind these experiments, and how the associated results should be interpreted. For all of these experiments (as with all of the experiments in this paper), we use the same node on a slurm cluster and distribute models across the 4 A100 GPUs on that node. For these experiments, we only use model parallelism.

Sampling configuration Similar to Hayes et al. (2025), after some initial tests with different sampling algorithm hyperparameters and example lengths, we use temperature $T = 1$ and top-k sampling with $k = 40$. We believe that setting $T = 1$ makes the most sense for studying memorization, as this reflects the LLM's base probability distribution. We clip that distribution to the top-40 tokens to only consider sequences of higher-probability sampled tokens.

For these experiments, we use 100-token-length examples $\boldsymbol{z}$, with a prefix (prompt) length of 50 tokens and a suffix (target generation) length of 50 tokens. We pick 100 tokens and a 50/50 split because this is accepted in the literature as a reasonable minimum for being confident that extracted sequences are reflective of memorization (Carlini et al., 2023b; 2021; Lee et al., 2022; Hayes et al., 2025). Some work uses shorter sequences (e.g., 64 tokens and a 32/32 token split (Biderman et al., 2023b)). While there is a reasonable argument to be made that such shorter sequences are still sufficiently long to be reflective of memorization, we opt for the more-standard choice of slightly longer prefixes and suffixes. Note that because our implementation of (n, p)-discoverable extraction makes computing $\log$ probabilities a post-processing operation on the output model logits (see Appendix A.2, in particular Listing 2), we also collect results for greedy-sampled discoverable extraction (Lee et al., 2022; Carlini et al., 2021; Nasr et al., 2023) by setting $T = k = 1$.

For the plots that follow, we are examining the 50/50 prefix/suffix regime (with some notable exceptions in the discussion section, see Appendix H).

Sampling procedure. To identify regions of memorization, we take the following "panning for gold" approach. For a given book, we start at the beginning of the text file in `Books3`. We sample a chunk of text that is sufficiently long to contain 100 tokens of corresponding tokenized text: to do this, we take a chunk of 800 characters, tokenize, take the first 100 tokens of the resulting tokenized sequence as the example and discard the rest. (Each 100 token sequence is typically, but not always, reflective of 300-400 characters of text. Rare words and formatting make this vary considerably. This is why we use a character-chunk size of 800—to make sure we have sufficient head-room to always end up with a 100 token sequence, even if this is generally wasteful in the average case in terms of discarded tokens.) We then shift 10 *characters* forward in the book text and repeat this process; we do this for the entire length of the book, which results in approximately one example every 10 characters, i.e., `len(book_characters)`/10 total examples $\boldsymbol{z}$, for which we compute $p_{\boldsymbol{z}}$. For example, *The Great Gatsby* has $270,870$ characters, which results in roughly $27,000$ examples, for which we compute the $p_{\boldsymbol{z}}$ quantity from (n, p)-discoverable extraction. (See Section 2 and Appendix A.2.)

This means, of course, that the 100-token examples overlap significantly. (Since generally speaking, 100 tokens covers 300-400 characters, shifting only 10 characters means there is a lot of overlap.) The point of doing this is that we don't know how different models were trained; *a priori*, it's not clear exactly where we should begin examples (or where they should end). With this approach, for extractable content, we expect to surface high-probability *regions* of memorized content, which we can then explore in more detail and more precisely in follow-up experiments. We picked a sliding window of 10 characters by running initial experiments with 1-, 10-, 50-, and 100-character windows. 10 characters exhibited an insignificant loss in extraction signal (at $10\times$ cheaper cost than 1 character). We intend to investigate reducing this cost further in the future.

We do *not* expect this approach to result in high per-book extraction rates. This is because, even though this approach *should* surface memorization hotspots, it will do so in a worst-case fashion. (This is precisely our goal, not deriving some procedure that maximizes the average extraction rate.) Since we are *indiscriminately* sliding the window 10 *characters* each time, we expect that many prompts will begin in "odd" locations—locations with "broken" or otherwise "strange" tokenizations, for which we expect to yield near-zero probability of extraction. For an intuition, consider this fake example. Imagine starting a prompt on `"ion"` instead of `"tion"` for `"determination"`, or any other word that ends in `"tion"`. The token for the word `"ion"` is likely/reasonably very different in the embedded token space from the token for `"tion"`, a very common suffix. Starting on `"ion"` (for a sentence containing a word that is something like `"determination"`, but with this context cropped out) should mean the probability of the continuing tokenized sequence should be low when run through the model.[7] That is true even of specialized names that might otherwise seem unique. Thus, while certain words may be more likely to follow `"Harry Potter"` in observed sentences, the same words would presumably not follow if the token begins with `"otter"` instead of `"Potter"`.

In summary, because many of the generated sequences should have near-zero extraction probability—due not only to the "broken" tokenization problem discussed above, but also because memorization (on average) is relatively rare in high-quality models, like those we study here—we expect this procedure to yield a relatively low average, book-specific extraction rates. We expect this even in books that contain substantial amounts of memorized content.

Nevertheless, in the subsections that follow, we show per-book extraction rates across the different models we test with this example-sampling procedure. We find this useful to get a very rough sense of the relative number of per-book memorization hot-spots, according to specific models. We also provide this long explanation because, for LLAMA 3.1 70B on some (very popular) books, we observe (unexpectedly) enormous extraction rates, which we will also revisit in the discussion in Appendix H), and which we allude to in Section 4, where we show some of the results that exhibit this.

Models tested with this procedure. We include results for 10 models: PYTHIA 12B (Biderman et al., 2023b), PHI 4 (a 14B model) (Abdin et al., 2024), LLAMA 1 13B, LLAMA 1 65B (Touvron et al., 2023a), LLAMA 2 70B (Touvron et al., 2023b), LLAMA 3 70B, LLAMA 3.1 70B (Grattafiori et al., 2024), DEEPSEEK V1 67B (DeepSeek-AI et al., 2024), QWEN 2.5 72B (Qwen et al., 2025), and GEMMA 2 27B (Team et al., 2024).

We pick these models for the following reasons: we want to include (the largest) PYTHIA model as a baseline, since we know for certain that these models were trained on The Pile (Gao et al., 2020) (which contains `Books3`), based on EleutherAI's extensive documentation. We also want to compare on a model that (should not) have been trained on `Books3`, based on its described training procedure. We pick PHI 4 for these reasons, since Abdin et al. (2024) was trained predominantly on synthetic data: "Synthetic data constitutes the bulk of the training data for phi-4 and is generated using a diverse array of techniques, including multi-agent prompting, self-revision workflows, and instruction reversal." (PHI 4 was also trained on "web content, *licensed* books, and code repositories to extract seeds for the synthetic data pipeline," emphasis added.) Further, PHI 4 is 14B parameters, which is a similar size to PYTHIA 12B, which facilitates cross-model comparisons of memorization. (This is because the amount of memorization, reasonably, also varies according to model size; see also Carlini et al. (2023b).) We include LLAMA 1 13B because it is (again) a similar size to PYTHIA 12B and PHI 4, and because we know the LLAMA 1 family was trained on `Books3` (Touvron et al., 2023a). However, we're most interested in testing larger models, since they tend to exhibit more memorization. We include 6 models that are around 70B. We include GEMMA 2 27B since it is the largest available open-weight continuation-style text model from Google (which otherwise would not be represented in our experiments). In the future, we will expand this to a much wider set of models. We are particularly interested in expanding to larger models trained by a wider range of organizations and companies.

Visualizations in the subsections that follow. As discussed above, for each book we include a per-book, cross-model comparison of average extraction rate. This is just to provide a high-level summary of relative extraction across these models. Since we are using the same underlying population of samples, these are valid comparisons, even if we don't expect the overall extraction rates to be high.

We then provide a table with the exact extraction rates that we plot: the greedy-sampled discoverable extraction rate, the probabilistic discoverable extraction rate (for the modest setting of $p_{\boldsymbol{z}} \geq 0.01\%$), the maximum-possible probabilistic discoverable extraction rate. (The maximum rate captures both valid instances of memorization, and for very large n, cases

[7]For example, common suffixes like `"tion"` tend to occupy clusters in embedding space because they appear in similar contexts. Fragments or rare (sub)words like `"ion"` tend to be more scattered/ not semantically coherent units. Similarly, this is true `"otter"` and `"Potter"`.

that could be argued to not necessarily be reflective entirely of "meaningful" memorization. We set aside the debate about what is "meaningful" to future work.) In this table, we also include the runtime of the experiment (using 4 A100s). We also include plots over the distribution of examples $\boldsymbol{z}$ whose $p_{\boldsymbol{z}} \geq 0.01\%$. These distributions show the counts of examples (per model, for each book) that have at least $p_{\boldsymbol{z}} \geq 0.01\%$.

Lastly, for some books, we include heatmaps that show the worst-case extraction by location (by character) in the book, in order to visualize where (if any) memorization "hot-spots" occur. That is, for a given character, for the 100 token sequences in which that character is included, we show the highest possible extraction probability (associated with a given 100-token sequence). Again, the point of this aligns with our overarching goal: visualizing memorization "hot-spots," for which this type of worst-case visualization is useful for eliciting how high-probability extraction regions ramp up and ramp down within the book. We don't include these heatmaps if they're mostly empty—i.e., don't contain several "hot-spots." We also omit heatmaps if the only "hot-spots" are at the very beginning or end of a book, as these (almost always) reflect extraction of copyright notices, publisher addresses, etc.

Tracking down longer extracted sequences. The results we obtain from these experiments—the probabilities of extraction $p_{\boldsymbol{z}}$ and the associated metadata we save (i.e., the position, in characters, of where each example is located within the book)—is useful for future experiments. Notably, by visualizing memorization "hot-spots," we can use this information to try to identify longer "runs" of memorized text.

The intuition is as follows. We take the full list of $p_{\boldsymbol{z}}$ for all examples for a given book and model experiment, along with their start position (character) in the book) and end position. (They are sorted in order by start position.) We filter the list to only contain especially high probability elements (e.g., $p_{\boldsymbol{z}} \geq 10\%$), and then iterate through this filtered list and merge together elements that overlap in terms of their position (taking the minimum start position as the overall start, and the maximum end position as the overall end). From here, we can then expect runs of regions in the book that have extractable examples with $p_{\boldsymbol{z}}$ greater than the threshold we filtered by (in this example, 10%). We can then run extraction experiments on those larger runs, using the start and end locations that we've identified through merging (in character space) adjacent examples that are high-probability.

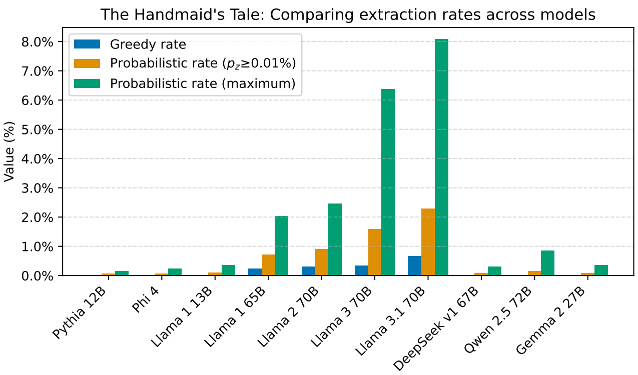
We attempted this simple methodology for *Harry Potter* (Rowling, 1998) and *1984* (Orwell, 1949) for LLAMA 3.1 70B, given the extent to which that model memorized those books. There is a lot more work to do in order to try to extract longer sequences, but we provide two examples of successes of this process in Appendix G. One of the examples is 300 tokens long, with a 25-token prefix and a 275-token suffix (Appendix G).

Estimating memorization of a whole book. The above identification of high-probability runs can also be used to estimate the proportion of a book that is extractable (with a minimum probability for $p_{\boldsymbol{z}}$). We show instances of the following in Figure 3 (right) for *Harry Potter* (Rowling, 1998) and *Sandman Slim* (Kadrey, 2009). Again, we pick a minimum threshold we are interested in for $p_{\boldsymbol{z}}$ (in Figure 3, we do this three times, picking $p_{\boldsymbol{z}} \in \{1\%, 10\%50\%, 75\%\}$). We filter the whole (sorted by start position) list of $p_{\boldsymbol{z}}$ to only contain examples whose extraction probability is greater than or equal to that threshold, and we merge (in terms of there location) adjacent examples to find the minimum start position and maximum end position.

We then end up with a merged list of "runs" of longer sequences that contain examples where we extracted text with at least $p_{\boldsymbol{z}}$ probability. We can compute the length of these "run" (in terms of total characters), sum them all up, and then divide by the total number of characters in the whole book. This gives us an estimate of the proportion of the whole book (in characters) for which there exist examples $\boldsymbol{z}$ (in tokens) that can be extracted with at least probability $p_{\boldsymbol{z}}$. This doesn't mean we can extract the whole book in one go, or that we've identified the minimal set of examples to try to extract. This is just a procedure for estimating (in a simple way) the proportion of total memorized text.

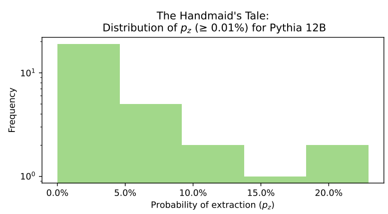
We defer further discussion to Appendix H.

F.1.1. *The Handmaid's Tale*, ATWOOD

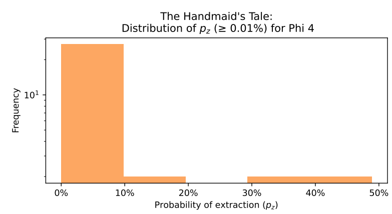


The Handmaid's Tale				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.02%	0.06%	0.15%	00:15:12.23
PHI 4	0.01%	0.06%	0.23%	00:26:39.96
LLAMA 1 13B	0.02%	0.09%	0.35%	00:16:32.97
LLAMA 1 65B	0.24%	0.71%	2.03%	00:56:07.73
LLAMA 2 70B	0.30%	0.90%	2.45%	00:56:03.78
LLAMA 3 70B	0.33%	1.59%	6.37%	01:08:44.77
LLAMA 3.1 70B	0.66%	2.28%	8.08%	01:08:14.35
DEEPSEEK V1 67B	0.01%	0.08%	0.30%	01:14:59.54
QWEN 2.5 72B	0.02%	0.15%	0.86%	01:14:45.41
GEMMA 2 27B	0.01%	0.07%	0.36%	01:14:34.15

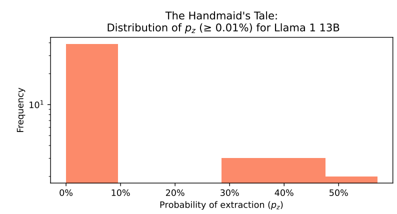
Figure 14: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



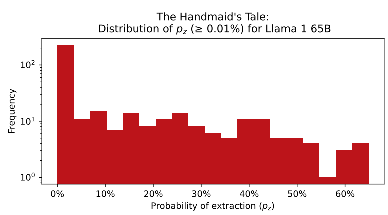
(a) PYTHIA 12B



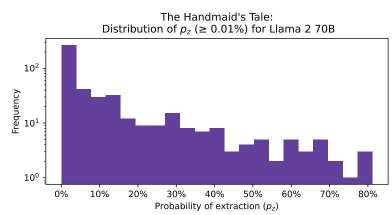
(b) PHI 4 (14B)



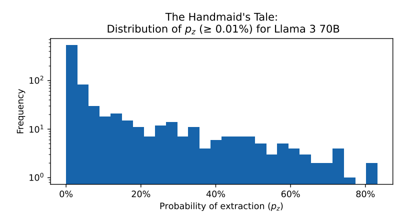
(c) LLAMA 1 13B



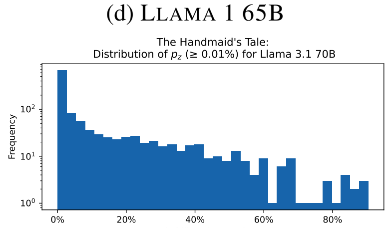
(d) LLAMA 1 65B



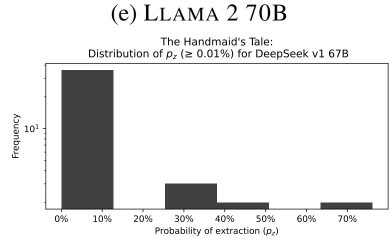
(e) LLAMA 2 70B



(f) LLAMA 3 70B

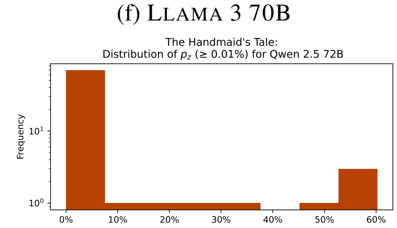


(g) LLAMA 3.1 70B

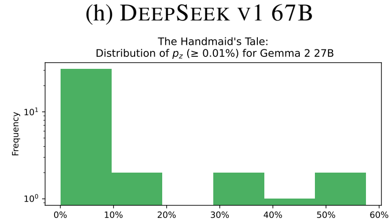


s

(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 15: *The Handmaid's Tale*, Atwood (1985): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

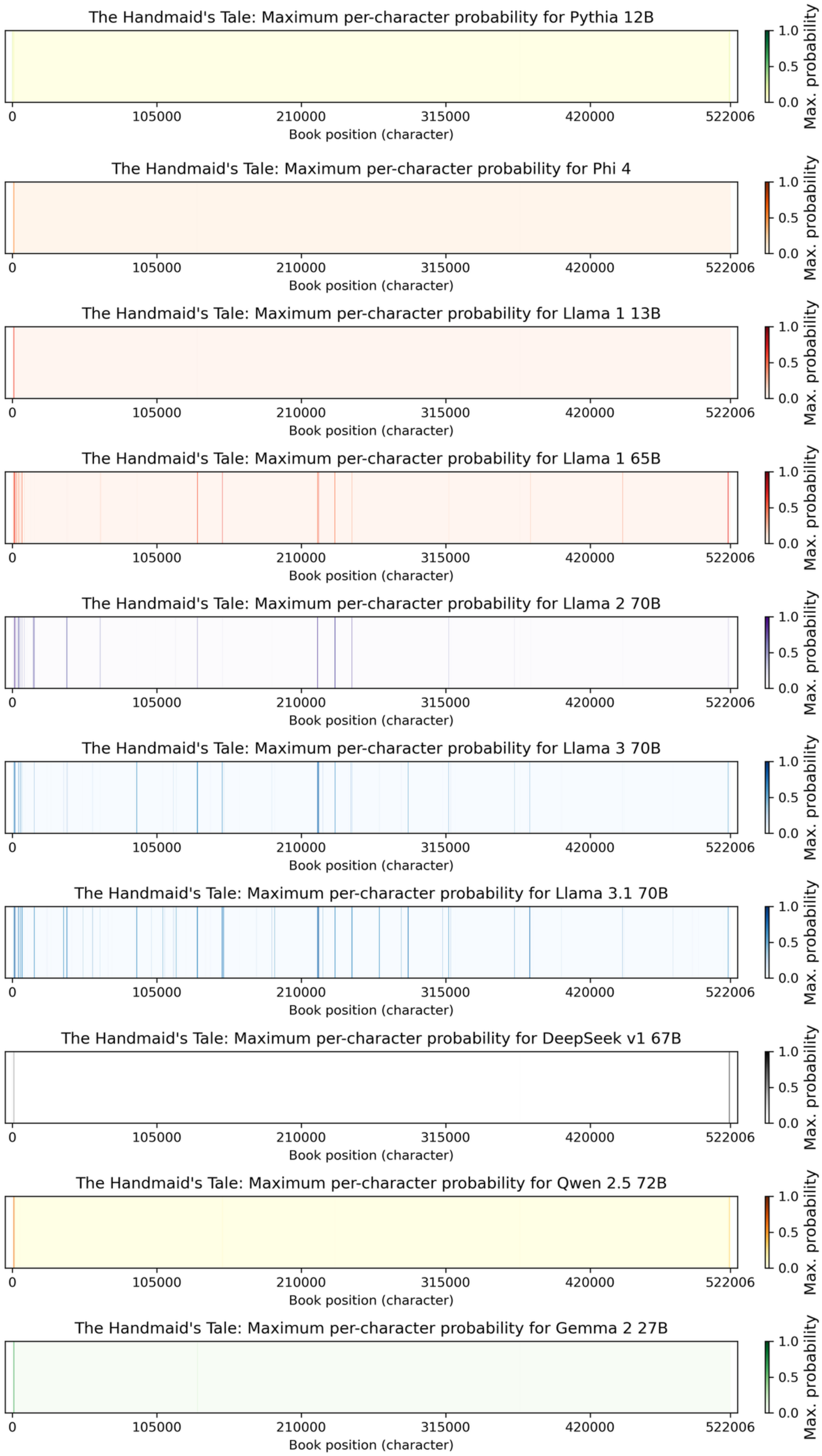
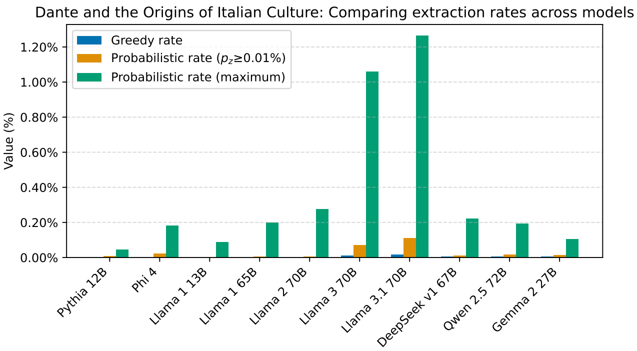


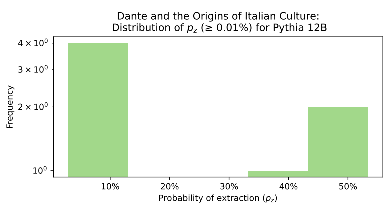
Figure 16: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.2. *Dante and the Origins of Italian Culture*, BAROLINI

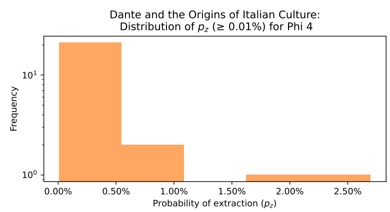


Dante and the Origins of Italian Culture				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.01%	0.04%	01:09:53.43
PHI 4	0.00%	0.02%	0.18%	02:18:03.72
LLAMA 1 13B	0.00%	0.00%	0.09%	00:50:57.34
LLAMA 1 65B	0.00%	0.00%	0.20%	02:20:26.40
LLAMA 2 70B	0.00%	0.01%	0.27%	02:06:17.05
LLAMA 3 70B	0.01%	0.07%	1.06%	02:32:48.51
LLAMA 3.1 70B	0.02%	0.11%	1.26%	02:32:13.24
DEEPSEEK V1 67B	0.00%	0.01%	0.22%	02:33:57.80
QWEN 2.5 72B	0.00%	0.02%	0.19%	02:52:29.34
GEMMA 2 27B	0.00%	0.01%	0.10%	02:38:54.09

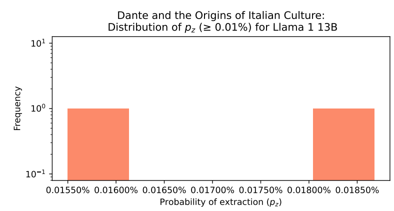
Figure 17: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



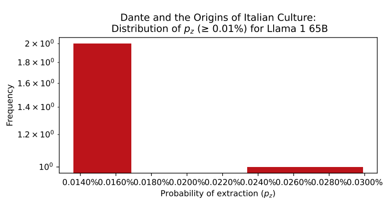
(a) PYTHIA 12B



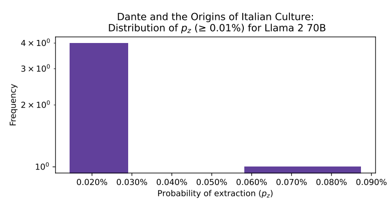
(b) PHI 4 (14B)



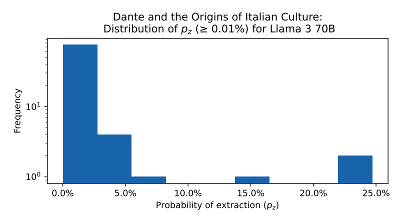
(c) LLAMA 1 13B



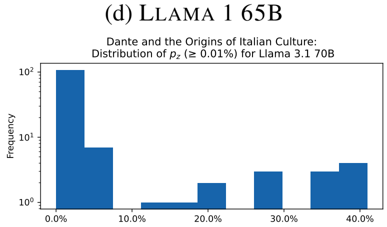
(d) LLAMA 1 65B



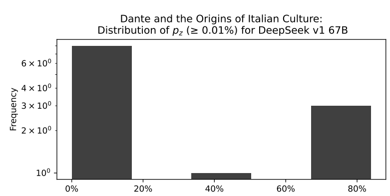
(e) LLAMA 2 70B



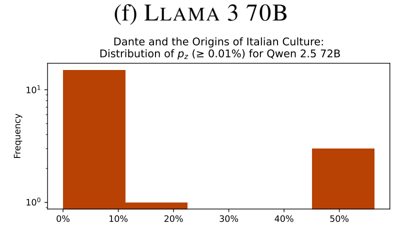
(f) LLAMA 3 70B



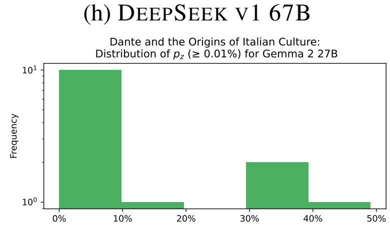
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



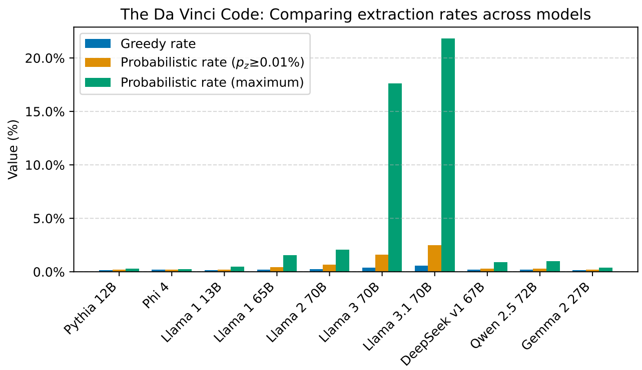
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

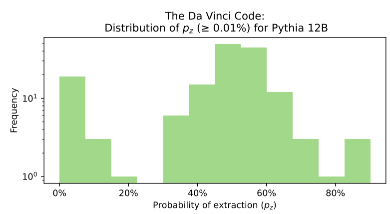
Figure 18: *Dante and the Origins of Italian Culture*, Barolini (2006): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Extracted examples are at the beginning and end of the book.

F.1.3. *The Da Vinci Code*, BROWN

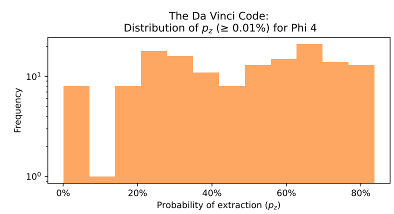


The Da Vinci Code				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.15%	0.19%	0.26%	00:25:41.35
PHI 4	0.16%	0.17%	0.22%	00:45:56.32
LLAMA 1 13B	0.16%	0.20%	0.47%	00:26:15.45
LLAMA 1 65B	0.21%	0.43%	1.56%	01:28:07.24
LLAMA 2 70B	0.24%	0.65%	2.06%	01:29:53.68
LLAMA 3 70B	0.39%	1.59%	17.61%	01:49:20.95
LLAMA 3.1 70B	0.54%	2.49%	21.80%	01:50:22.05
DEEPSEEK V1 67B	0.17%	0.27%	0.87%	01:55:44.93
QWEN 2.5 72B	0.17%	0.28%	0.97%	02:14:58.24
GEMMA 2 27B	0.16%	0.19%	0.39%	01:54:16.65

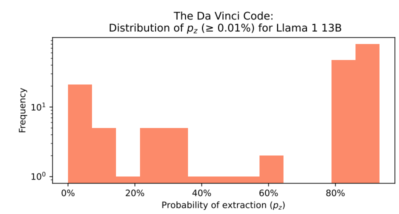
Figure 19: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



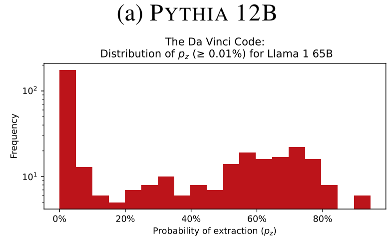
(a) PYTHIA 12B



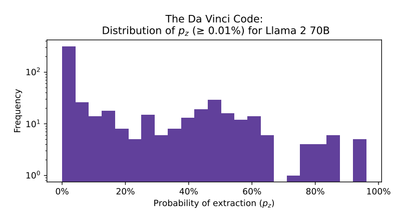
(b) PHI 4 (14B)



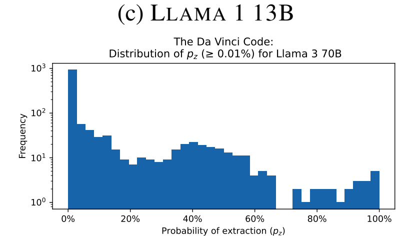
(c) LLAMA 1 13B



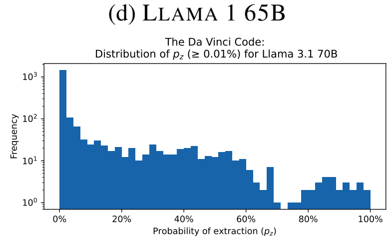
(d) LLAMA 1 65B



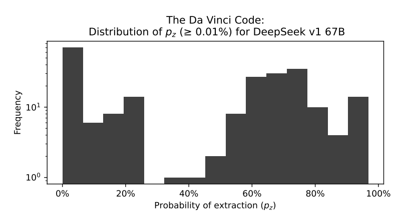
(e) LLAMA 2 70B



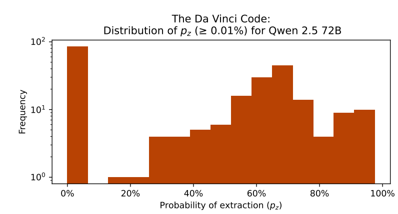
(f) LLAMA 3 70B



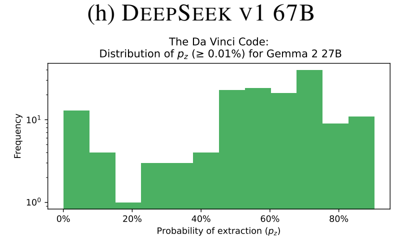
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 20: *The Da Vinci Code*, Brown (2003): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

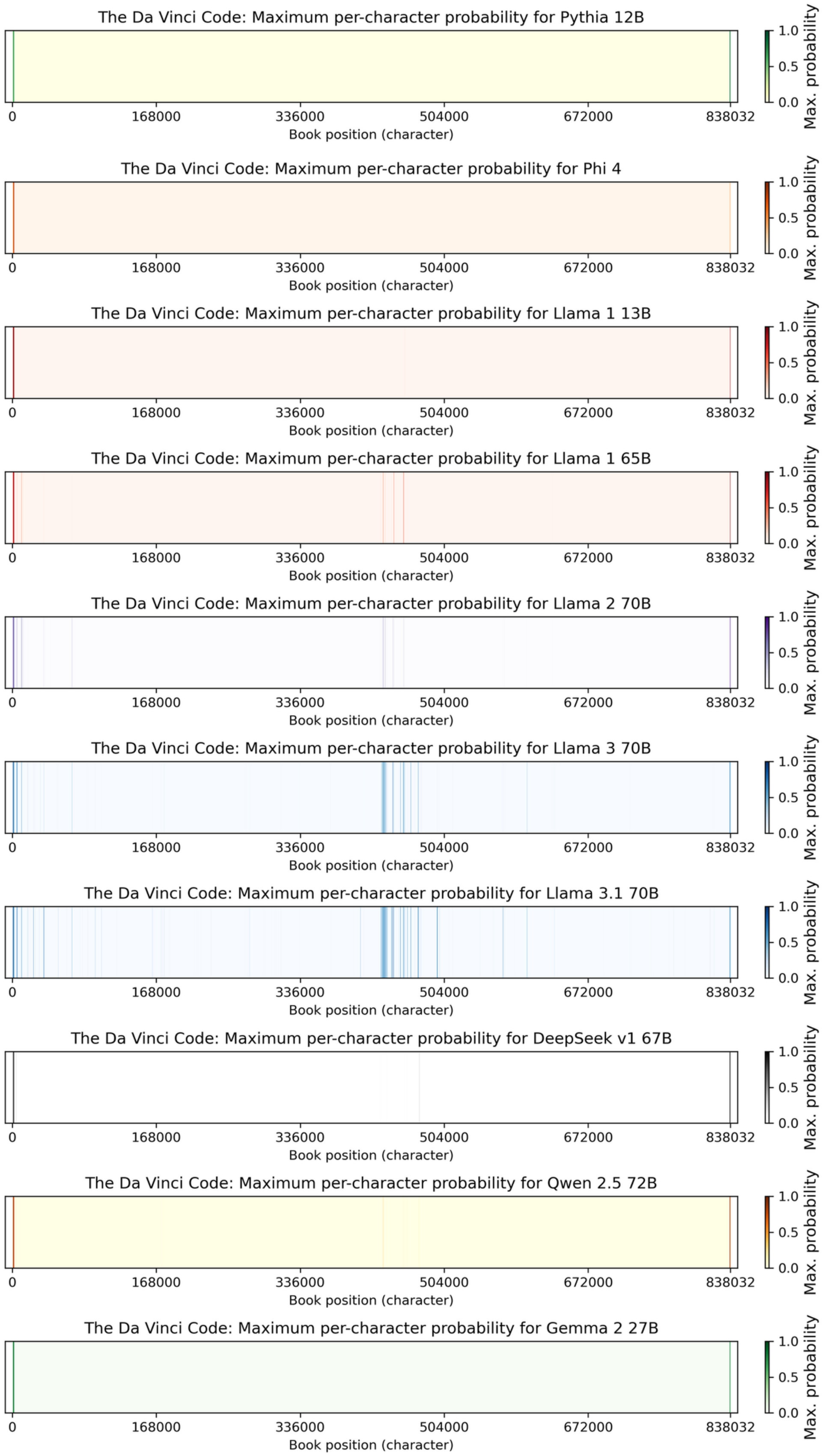
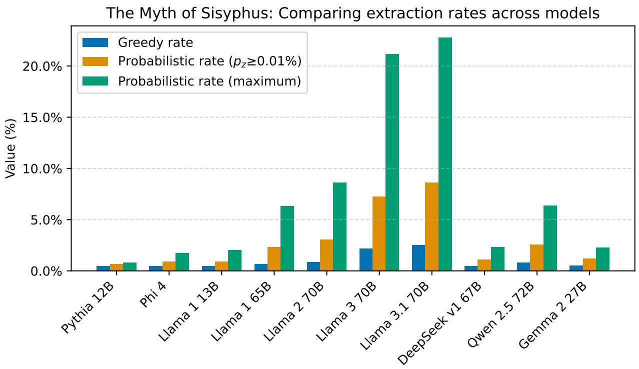


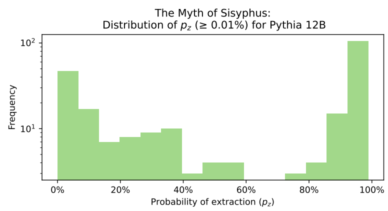
Figure 21: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.4. *The Myth of Sisyphus*, Camus

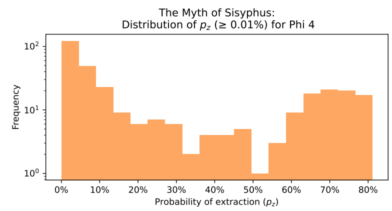


The Myth of Sisyphus				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
Pythia 12B	0.45%	0.66%	0.81%	00:10:47.48
Phi 4	0.47%	0.92%	1.76%	00:22:30.60
Llama 1 13B	0.48%	0.91%	2.03%	00:11:17.22
Llama 1 65B	0.68%	2.30%	6.30%	00:38:00.27
Llama 2 70B	0.86%	3.05%	8.62%	00:38:46.04
Llama 3 70B	2.15%	7.26%	21.17%	00:46:57.15
Llama 3.1 70B	2.50%	8.64%	22.78%	00:47:50.27
DeepSeek v1 67B	0.45%	1.10%	2.32%	00:56:29.95
Qwen 2.5 72B	0.79%	2.57%	6.38%	01:03:05.52
Gemma 2 27B	0.49%	1.18%	2.29%	00:56:54.18

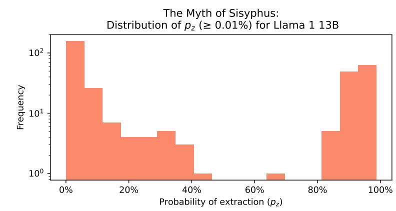
Figure 22: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



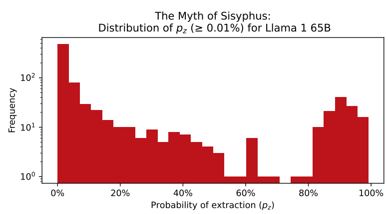
(a) Pythia 12B



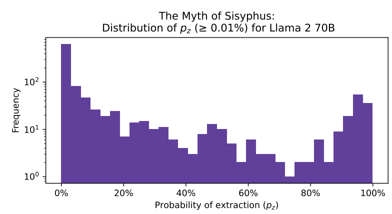
(b) Phi 4 (14B)



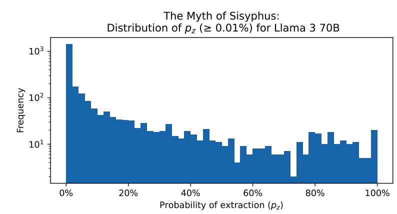
(c) Llama 1 13B



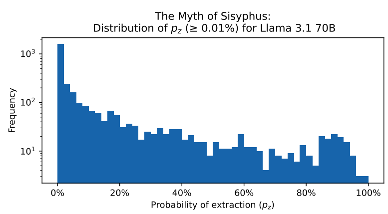
(d) Llama 1 65B



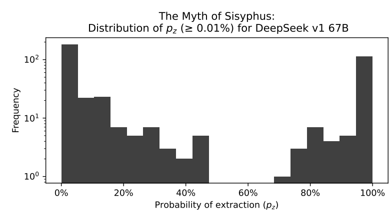
(e) Llama 2 70B



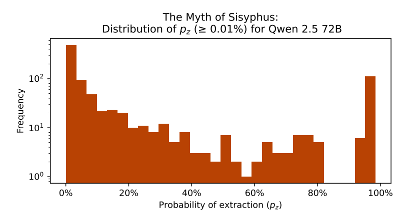
(f) Llama 3 70B



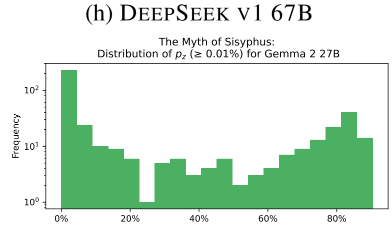
(g) Llama 3.1 70B



(h) DeepSeek v1 67B



(i) Qwen 2.5 72B



(j) Gemma 2 27B

Figure 23: *The Myth of Sisyphus*, Camus (1955): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

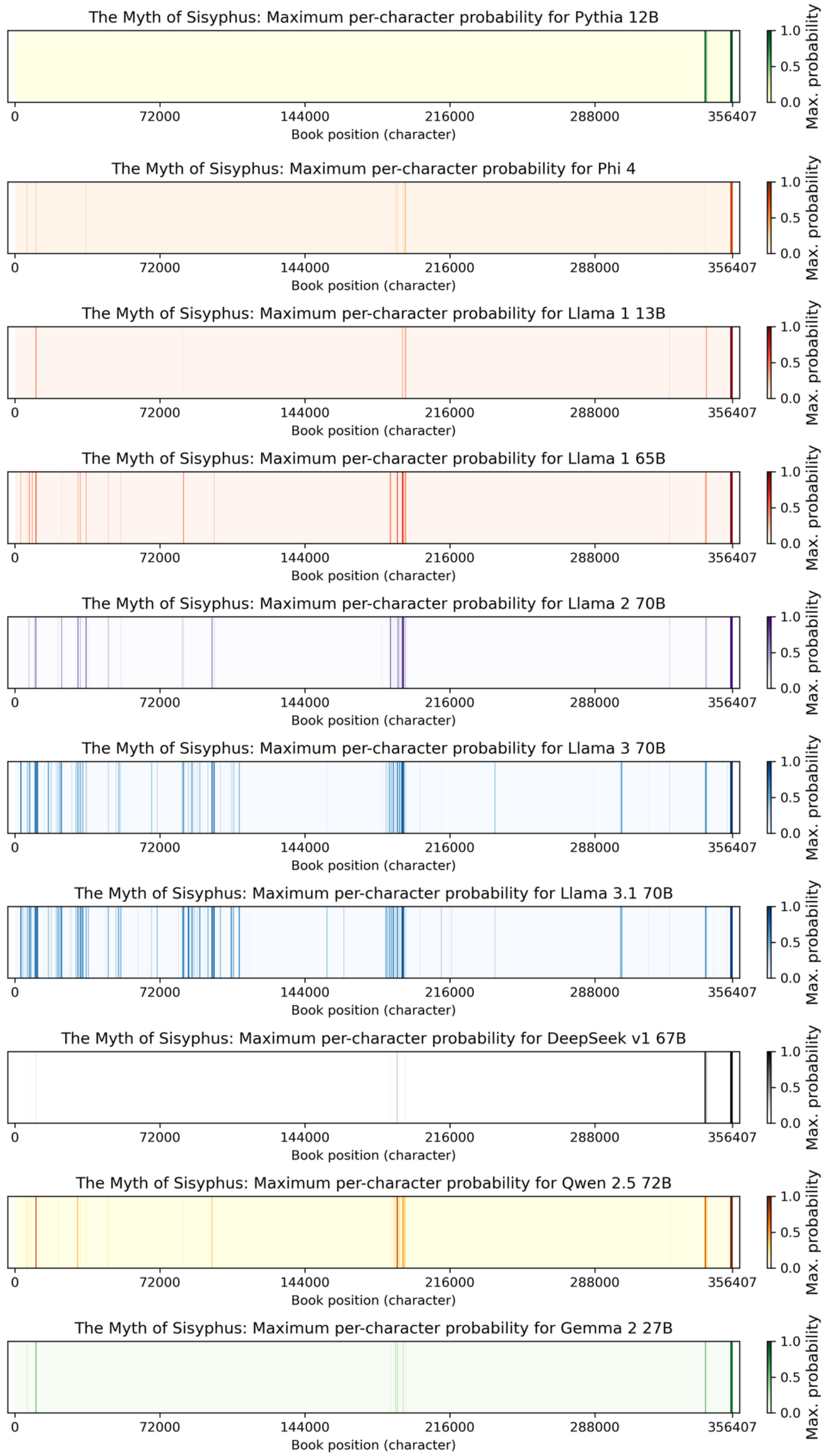
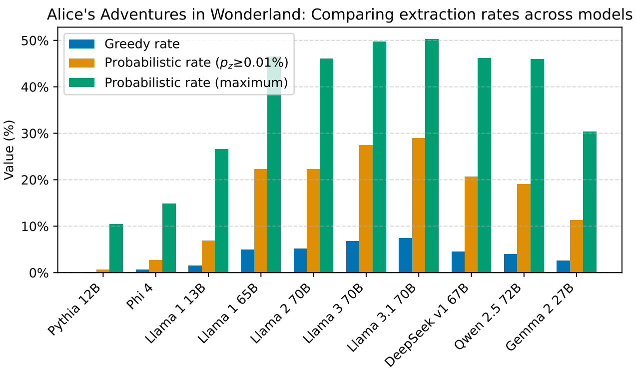


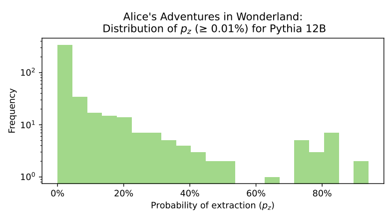
Figure 24: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.5. *Alice's Adventures in Wonderland*, CARROLL

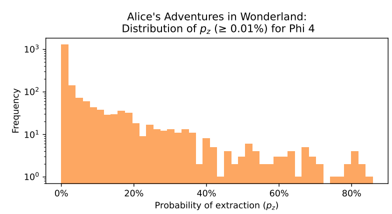


Alice's Adventures in Wonderland				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.16%	0.65%	10.40%	00:42:22.36
PHI 4	0.65%	2.73%	14.90%	01:24:06.58
LLAMA 1 13B	1.49%	6.90%	26.56%	00:32:19.81
LLAMA 1 65B	4.96%	22.24%	46.52%	01:26:03.29
LLAMA 2 70B	5.19%	22.33%	46.08%	01:17:43.11
LLAMA 3 70B	6.76%	27.41%	49.74%	01:34:00.43
LLAMA 3.1 70B	7.48%	28.95%	50.32%	01:35:36.94
DEEPSEEK V1 67B	4.52%	20.68%	46.19%	01:39:10.60
QWEN 2.5 72B	4.02%	19.00%	45.95%	01:52:45.29
GEMMA 2 27B	2.57%	11.26%	30.32%	01:36:46.76

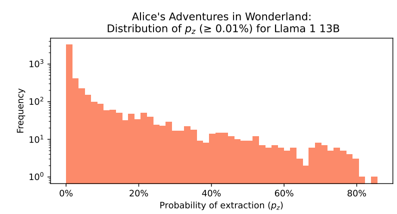
Figure 25: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



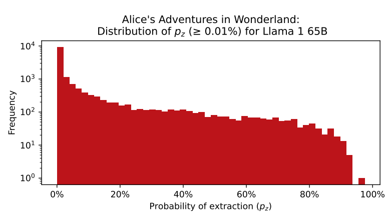
(a) PYTHIA 12B



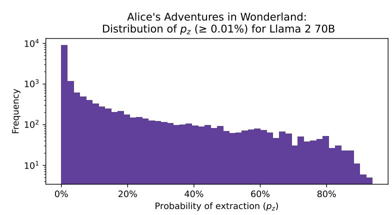
(b) PHI 4 (14B)



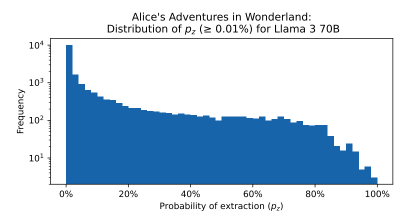
(c) LLAMA 1 13B



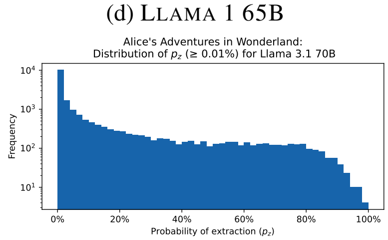
(d) LLAMA 1 65B



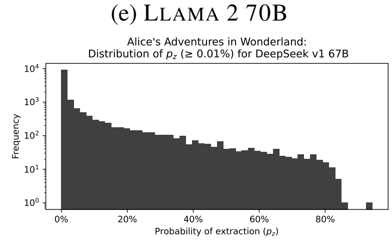
(e) LLAMA 2 70B



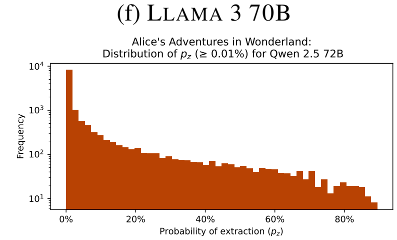
(f) LLAMA 3 70B



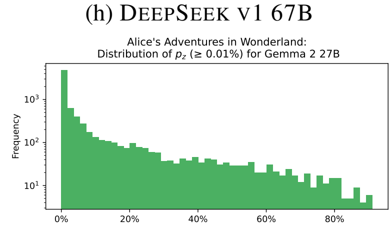
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 26: *Alice's Adventures in Wonderland*, Carroll (1865): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

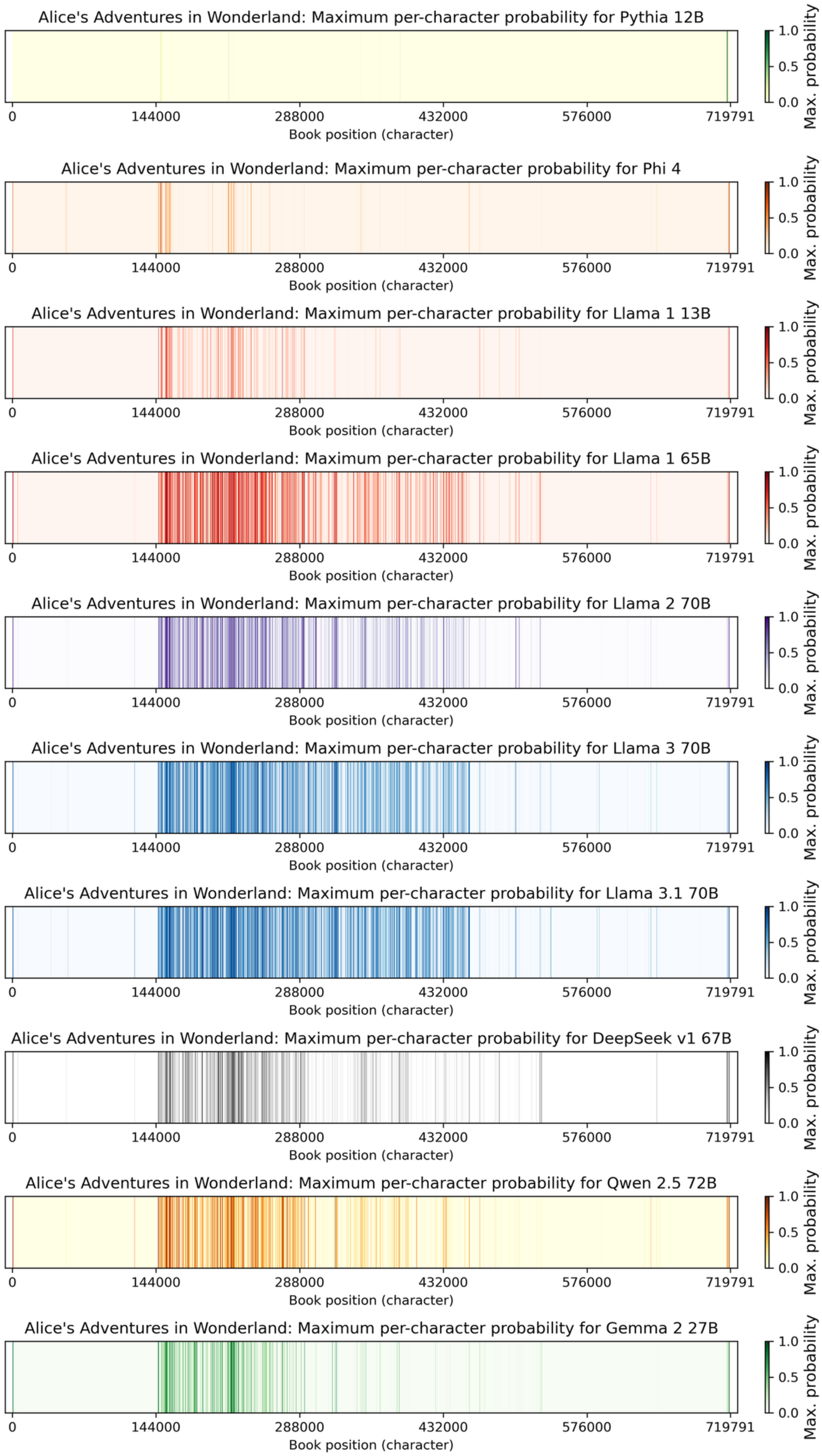
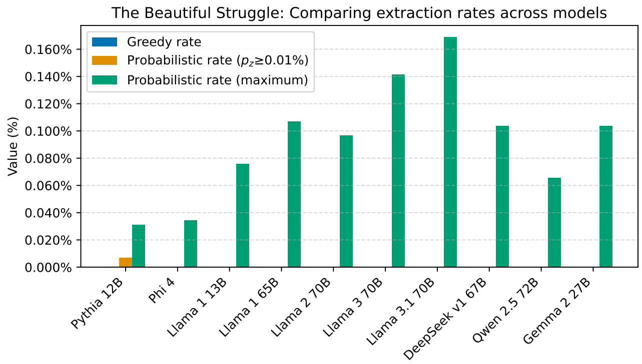


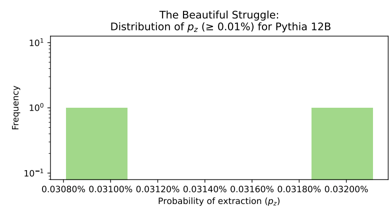
Figure 27: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.6. *The Beautiful Struggle*, COATES



The Beautiful Struggle				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.01%	0.03%	00:19:22.45
PHI 4	0.00%	0.00%	0.03%	00:39:00.05
LLAMA 1 13B	0.00%	0.00%	0.08%	00:13:44.55
LLAMA 1 65B	0.00%	0.00%	0.11%	00:37:14.86
LLAMA 2 70B	0.00%	0.00%	0.10%	00:31:25.61
LLAMA 3 70B	0.00%	0.00%	0.14%	00:38:11.48
LLAMA 3.1 70B	0.00%	0.00%	0.17%	00:39:49.66
DEEPSEEK V1 67B	0.00%	0.00%	0.10%	00:46:24.68
QWEN 2.5 72B	0.00%	0.00%	0.07%	00:53:11.56
GEMMA 2 27B	0.00%	0.00%	0.10%	00:42:27.90

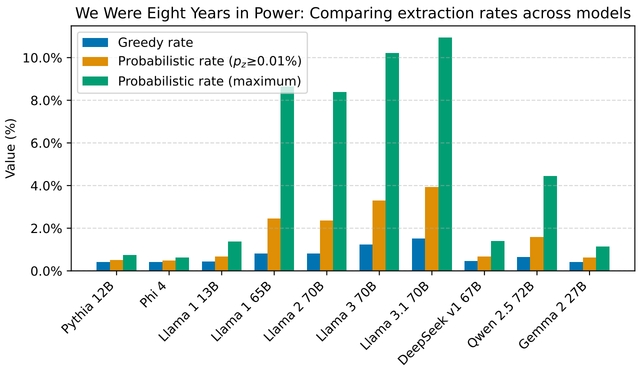
Figure 28: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



(a) PYTHIA 12B

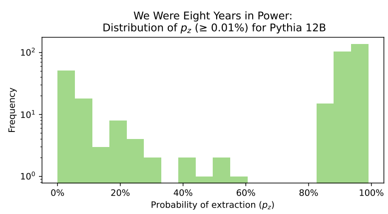
Figure 29: *The Beautiful Struggle*, Coates (2009): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. PYTHIA 12B is the only model that has any examples with $p_{\boldsymbol{z}} \geq 0.01\%$.

F.1.7. *We Were Eight Years in Power*, COATES

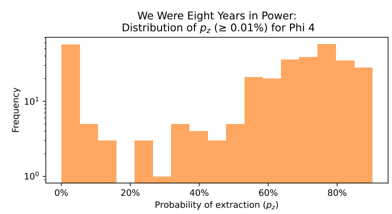


We Were Eight Years in Power				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.42%	0.51%	0.73%	00:21:20.90
PHI 4	0.41%	0.47%	0.62%	00:39:18.74
LLAMA 1 13B	0.43%	0.66%	1.38%	00:21:48.74
LLAMA 1 65B	0.80%	2.44%	8.64%	01:13:02.59
LLAMA 2 70B	0.80%	2.35%	8.38%	01:13:58.30
LLAMA 3 70B	1.24%	3.28%	10.20%	01:29:38.62
LLAMA 3.1 70B	1.51%	3.93%	10.94%	01:29:39.20
DEEPSEEK V1 67B	0.45%	0.68%	1.39%	01:36:56.81
QWEN 2.5 72B	0.65%	1.58%	4.45%	01:50:31.25
GEMMA 2 27B	0.41%	0.62%	1.14%	01:36:05.07

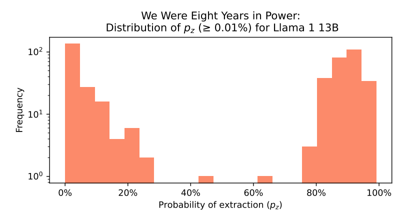
Figure 30: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



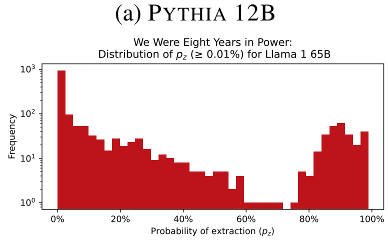
(a) PYTHIA 12B



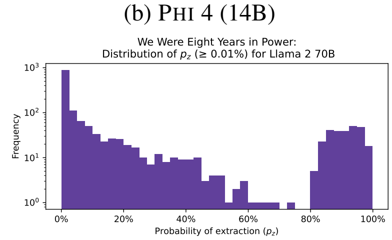
(b) PHI 4 (14B)



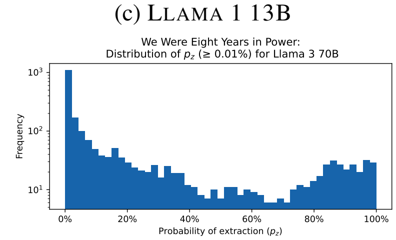
(c) LLAMA 1 13B



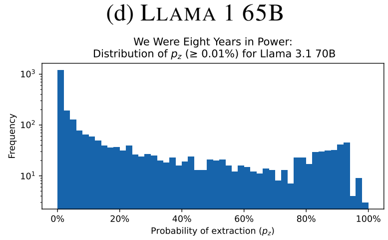
(d) LLAMA 1 65B



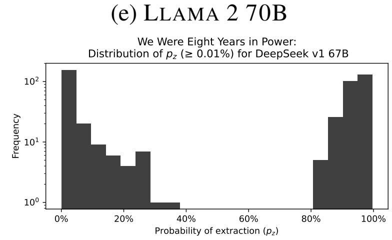
(e) LLAMA 2 70B



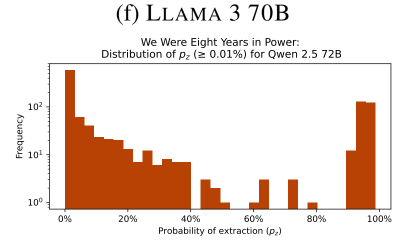
(f) LLAMA 3 70B



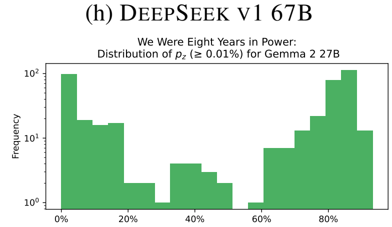
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 31: *We Were Eight Years in Power*, Coates (2017): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

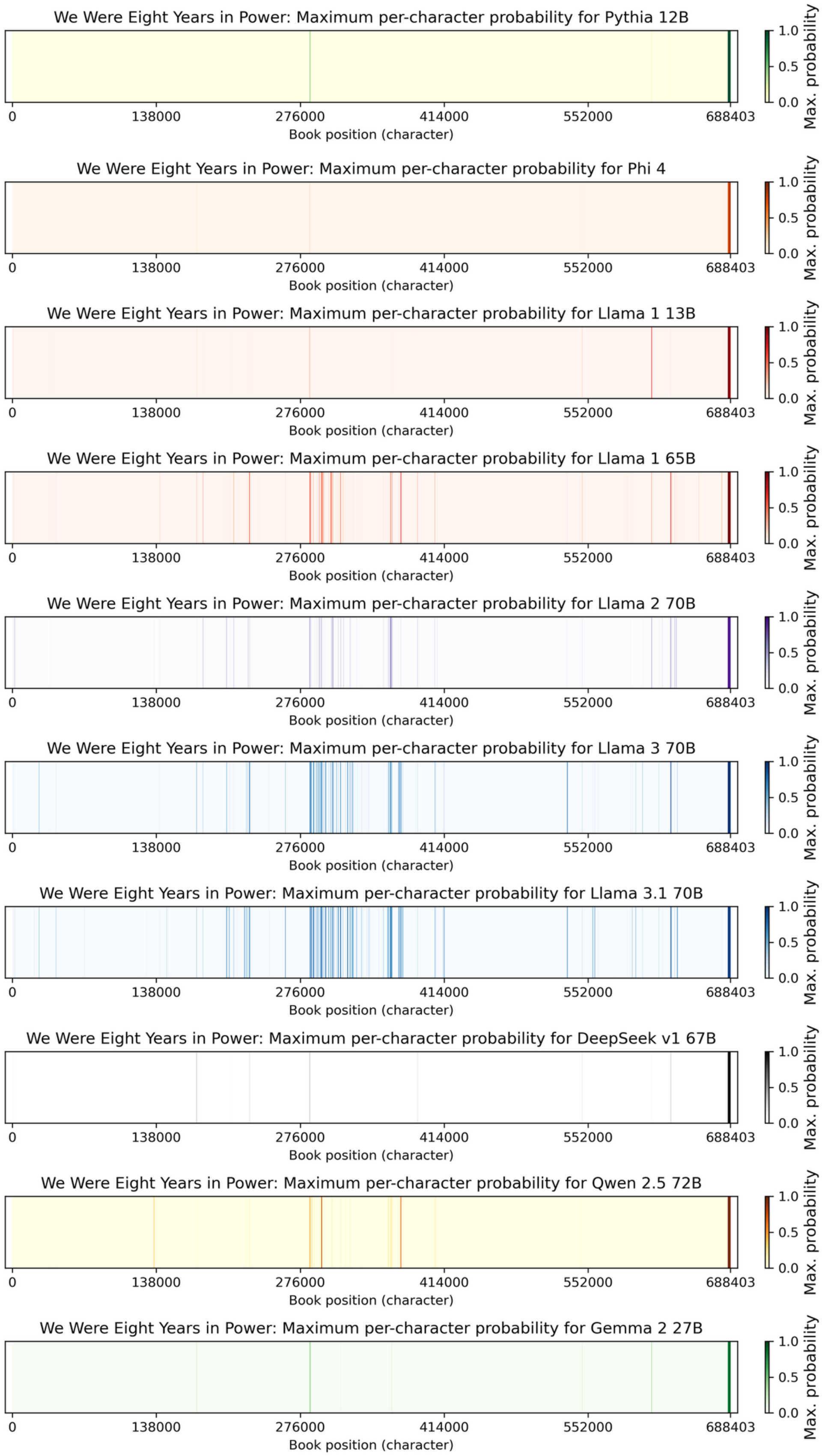
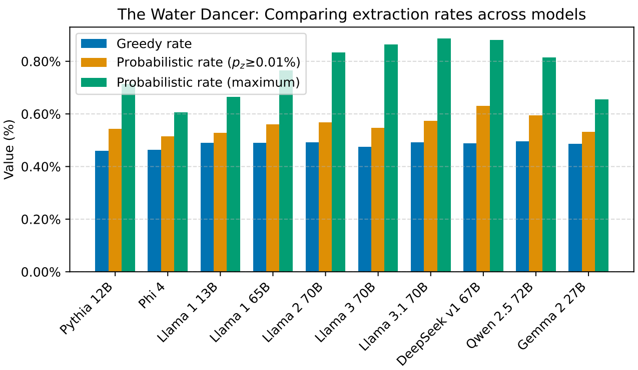


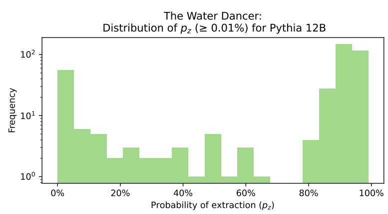
Figure 32: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.8. *The Water Dancer*, COATES

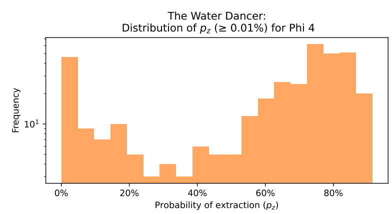


The Water Dancer				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.46%	0.54%	0.71%	00:21:44.11
PHI 4	0.46%	0.52%	0.61%	00:40:49.06
LLAMA 1 13B	0.49%	0.53%	0.67%	00:22:15.13
LLAMA 1 65B	0.49%	0.56%	0.76%	01:16:48.71
LLAMA 2 70B	0.49%	0.57%	0.83%	01:16:49.47
LLAMA 3 70B	0.47%	0.55%	0.86%	01:31:46.86
LLAMA 3.1 70B	0.49%	0.57%	0.89%	01:32:59.63
DEEPSEEK V1 67B	0.49%	0.63%	0.88%	01:42:56.38
QWEN 2.5 72B	0.50%	0.59%	0.81%	01:56:53.50
GEMMA 2 27B	0.49%	0.53%	0.66%	01:35:49.67

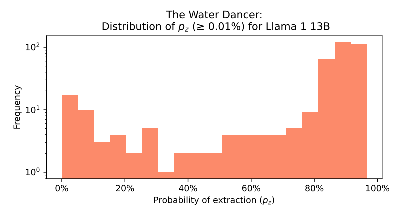
Figure 33: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



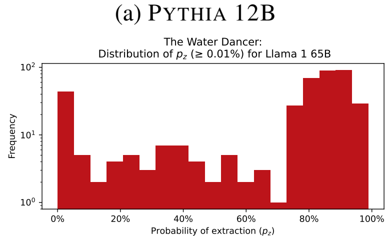
(a) PYTHIA 12B



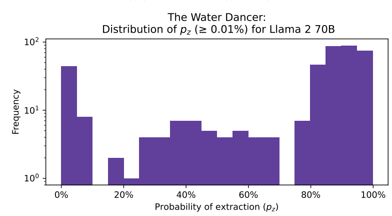
(b) PHI 4 (14B)



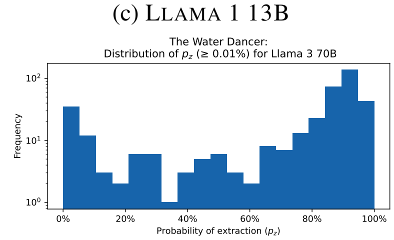
(c) LLAMA 1 13B



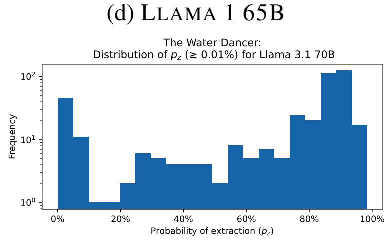
(d) LLAMA 1 65B



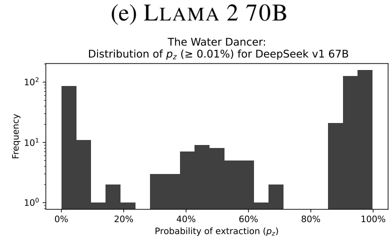
(e) LLAMA 2 70B



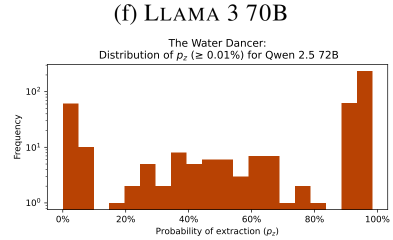
(f) LLAMA 3 70B



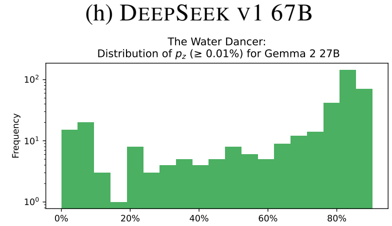
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



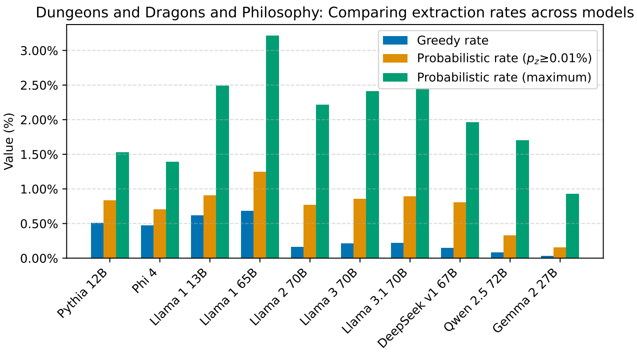
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

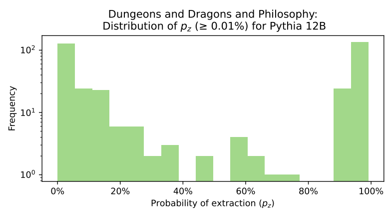
Figure 34: *The Water Dancer*, Coates (2019): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Extracted examples are at the beginning and end of the book.

F.1.9. *Dungeons and Dragons and Philosophy*, COGBURN

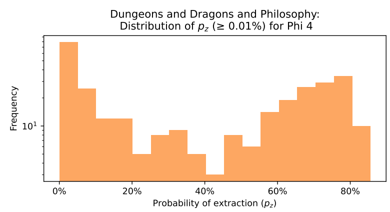


Dungeons and Dragons and Philosophy				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.51%	0.84%	1.53%	00:13:24.97
PHI 4	0.47%	0.71%	1.39%	00:25:07.75
LLAMA 1 13B	0.62%	0.91%	2.49%	00:13:22.00
LLAMA 1 65B	0.68%	1.25%	3.21%	00:46:01.07
LLAMA 2 70B	0.17%	0.77%	2.21%	01:12:18.74
LLAMA 3 70B	0.21%	0.86%	2.42%	01:27:54.71
LLAMA 3.1 70B	0.22%	0.89%	2.45%	01:29:09.97
DEEPSEEK V1 67B	0.15%	0.81%	1.96%	01:36:20.18
QWEN 2.5 72B	0.08%	0.33%	1.71%	01:43:50.77
GEMMA 2 27B	0.04%	0.15%	0.93%	01:27:37.69

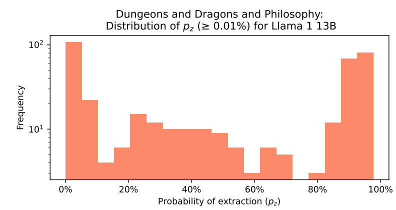
Figure 35: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



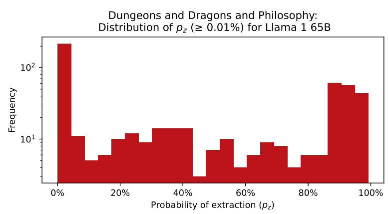
(a) PYTHIA 12B



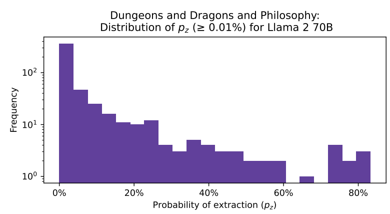
(b) PHI 4 (14B)



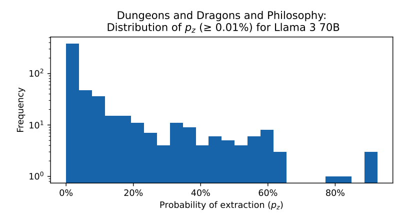
(c) LLAMA 1 13B



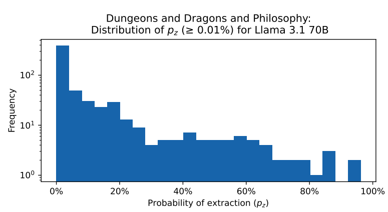
(d) LLAMA 1 65B



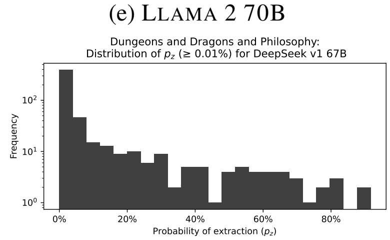
(e) LLAMA 2 70B



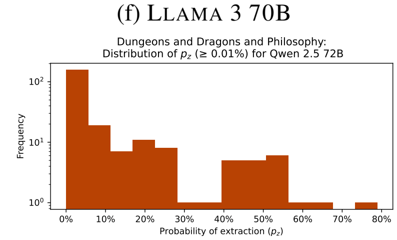
(f) LLAMA 3 70B



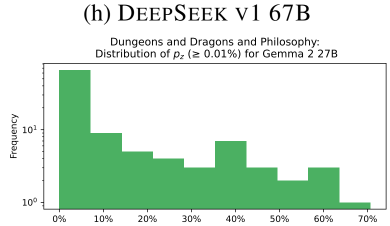
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 36: *Dungeons and Dragons and Philosophy*, Cogburn (2012): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

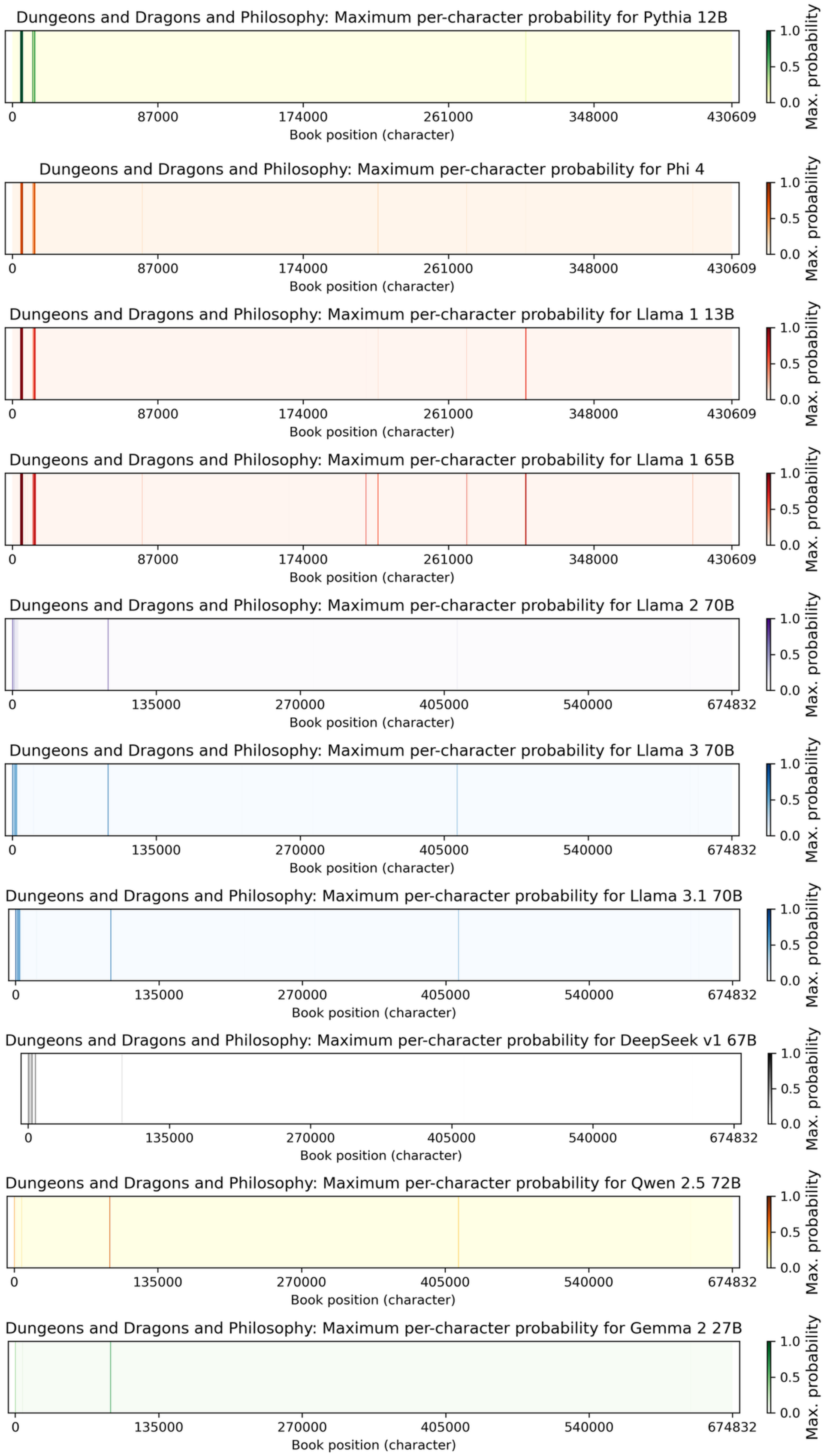
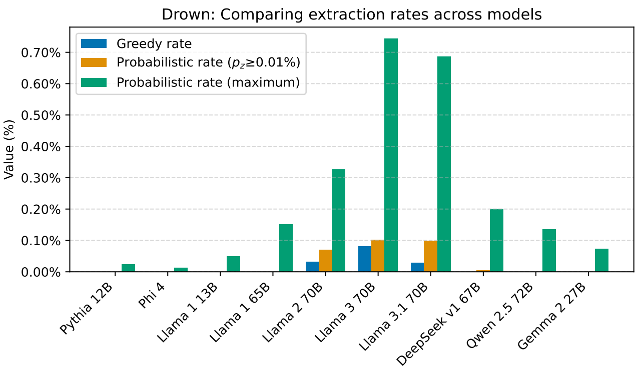


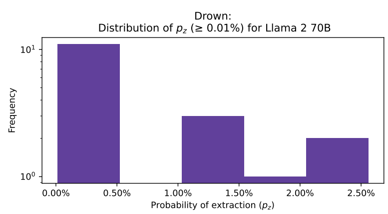
Figure 37: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.10. *Drown*, DÍAZ

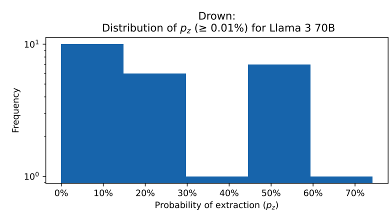


Drown				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.00%	0.02%	00:15:39.60
PHI 4	0.00%	0.00%	0.01%	00:31:49.62
LLAMA 1 13B	0.00%	0.00%	0.05%	00:11:21.27
LLAMA 1 65B	0.00%	0.00%	0.15%	00:30:33.16
LLAMA 2 70B	0.03%	0.07%	0.33%	00:26:45.87
LLAMA 3 70B	0.08%	0.10%	0.74%	00:33:39.80
LLAMA 3.1 70B	0.03%	0.10%	0.69%	00:32:19.88
DEEPSEEK V1 67B	0.00%	0.00%	0.20%	00:42:57.33
QWEN 2.5 72B	0.00%	0.00%	0.13%	00:42:54.30
GEMMA 2 27B	0.00%	0.00%	0.07%	00:34:08.02

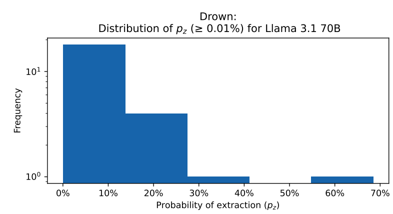
Figure 38: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n,p)-discoverable rate across different models.



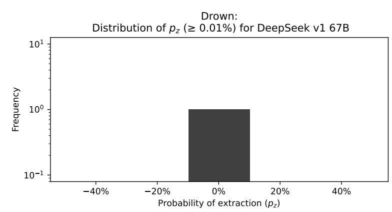
(a) LLAMA 2 70B



(b) LLAMA 3 70B



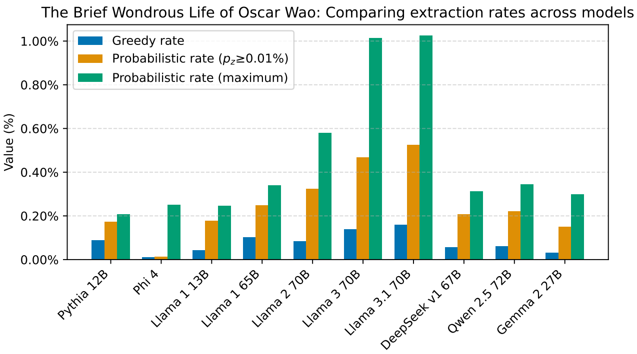
(c) LLAMA 3.1 70B



(d) DEEPSEEK V1 67B

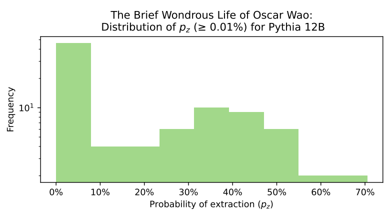
Figure 39: *Drown*, Díaz (1996): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Only showing models for which there exist any $p_{\boldsymbol{z}} \geq 0.01\%$.

F.1.11. *The Brief Wondrous Life of Oscar Wao*, Díaz

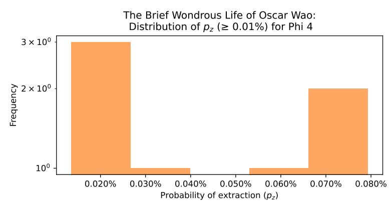


The Brief Wondrous Life of Oscar Wao				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.09%	0.17%	0.21%	00:17:12.42
PHI 4	0.01%	0.01%	0.25%	00:27:47.94
LLAMA 1 13B	0.04%	0.18%	0.25%	00:16:17.08
LLAMA 1 65B	0.10%	0.25%	0.34%	00:55:33.06
LLAMA 2 70B	0.08%	0.32%	0.58%	00:55:40.12
LLAMA 3 70B	0.14%	0.47%	1.01%	01:08:54.55
LLAMA 3.1 70B	0.16%	0.53%	1.02%	01:07:45.63
DEEPSEEK V1 67B	0.06%	0.21%	0.31%	01:12:30.63
QWEN 2.5 72B	0.06%	0.22%	0.34%	01:20:54.30
GEMMA 2 27B	0.03%	0.15%	0.30%	01:13:44.72

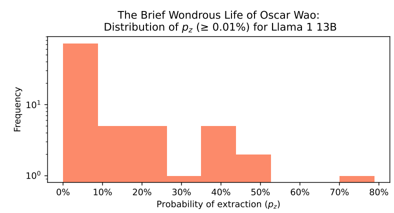
Figure 40: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



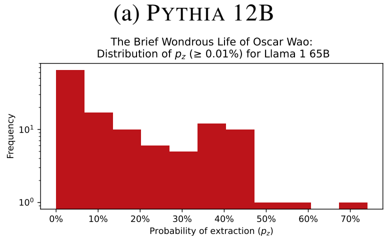
(a) PYTHIA 12B



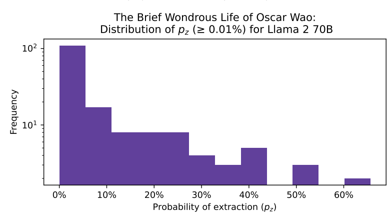
(b) PHI 4 (14B)



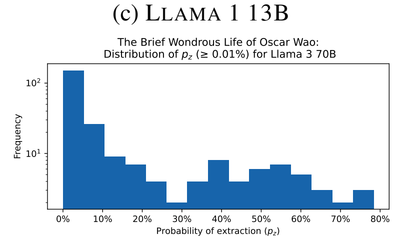
(c) LLAMA 1 13B



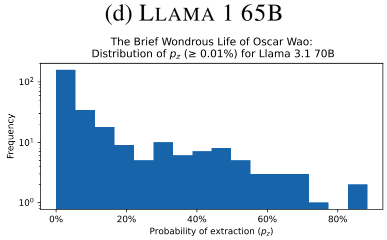
(d) LLAMA 1 65B



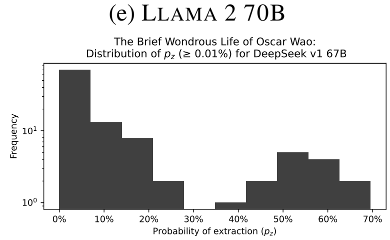
(e) LLAMA 2 70B



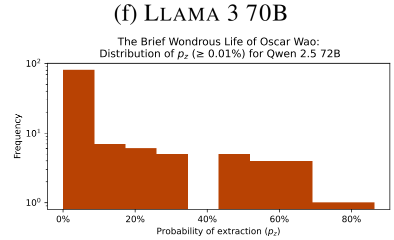
(f) LLAMA 3 70B



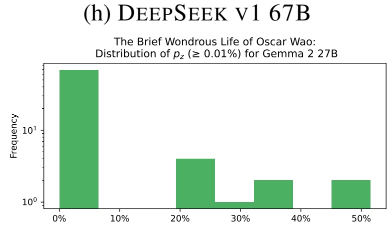
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 41: *The Brief Wondrous Life of Oscar Wao*, Díaz (2007): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

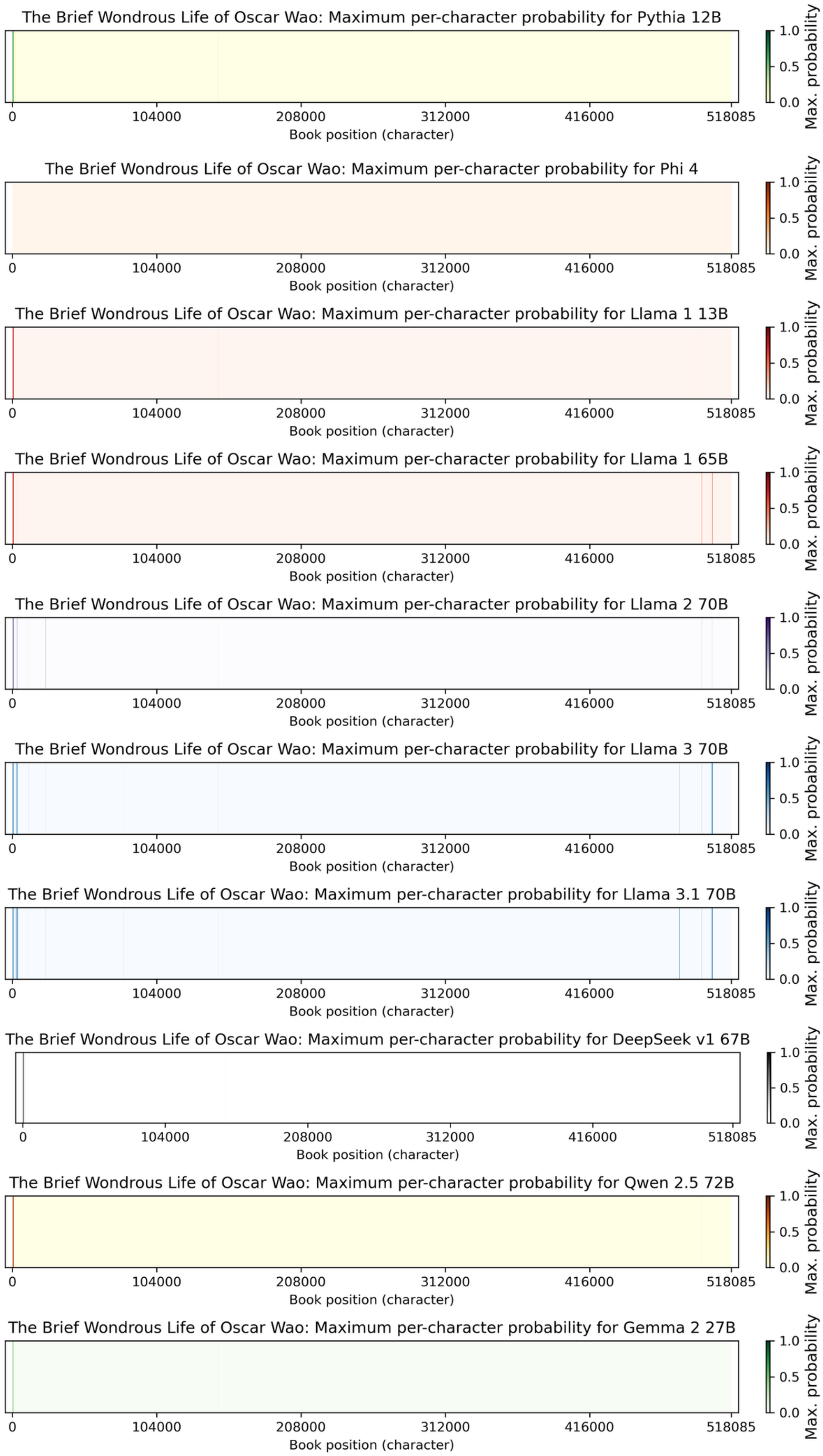
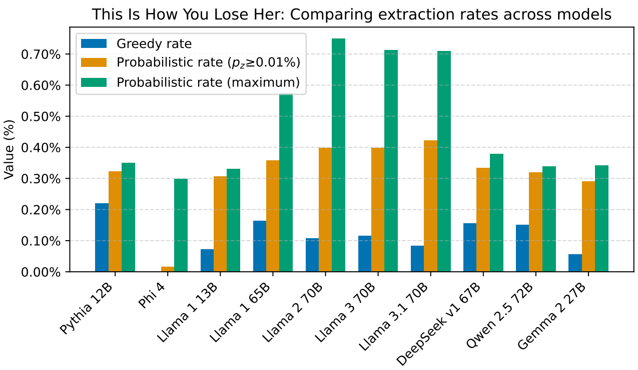


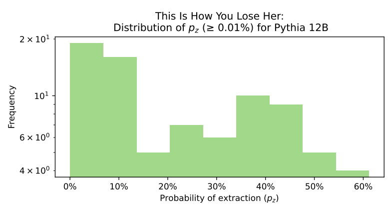
Figure 42: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.12. *This Is How You Lose Her*, DÍAZ

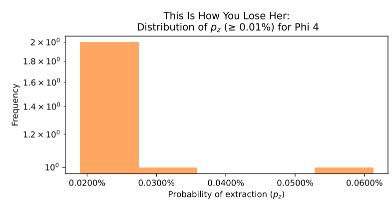


This Is How You Lose Her				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.22%	0.32%	0.35%	00:07:29.38
PHI 4	0.00%	0.02%	0.30%	00:12:51.69
LLAMA 1 13B	0.07%	0.31%	0.33%	00:07:57.88
LLAMA 1 65B	0.16%	0.36%	0.57%	00:27:38.38
LLAMA 2 70B	0.11%	0.40%	0.75%	00:27:25.49
LLAMA 3 70B	0.12%	0.40%	0.71%	00:33:17.95
LLAMA 3.1 70B	0.08%	0.42%	0.71%	00:33:24.74
DEEPSEEK V1 67B	0.16%	0.33%	0.38%	00:41:50.38
QWEN 2.5 72B	0.15%	0.32%	0.34%	00:37:56.86
GEMMA 2 27B	0.06%	0.29%	0.34%	00:38:33.11

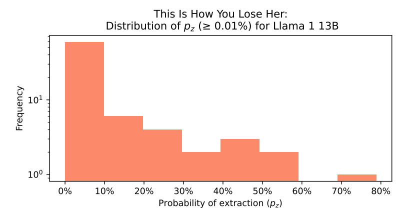
Figure 43: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



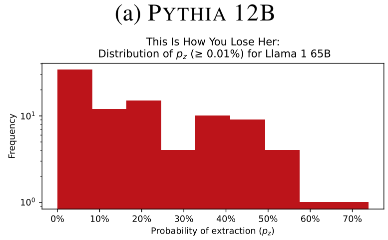
(a) PYTHIA 12B



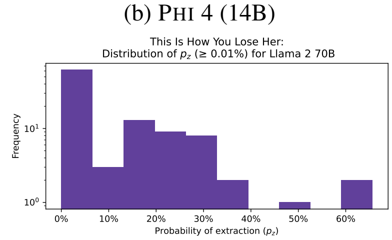
(b) PHI 4 (14B)



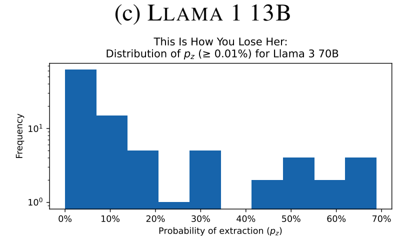
(c) LLAMA 1 13B



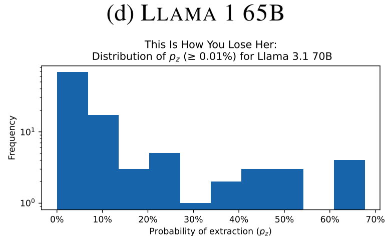
(d) LLAMA 1 65B



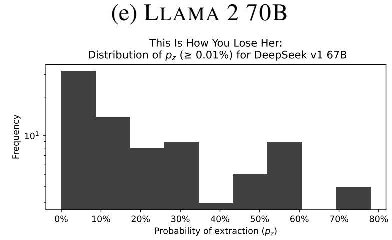
(e) LLAMA 2 70B



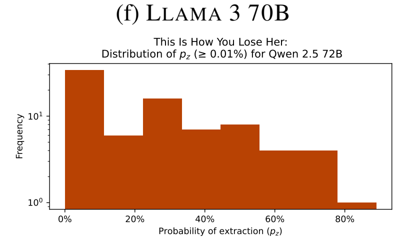
(f) LLAMA 3 70B



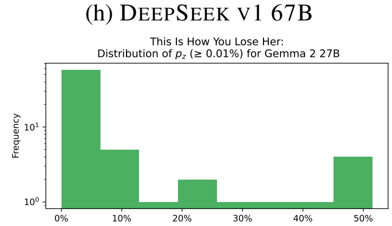
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 44: *This Is How You Lose Her*, Díaz (2012): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

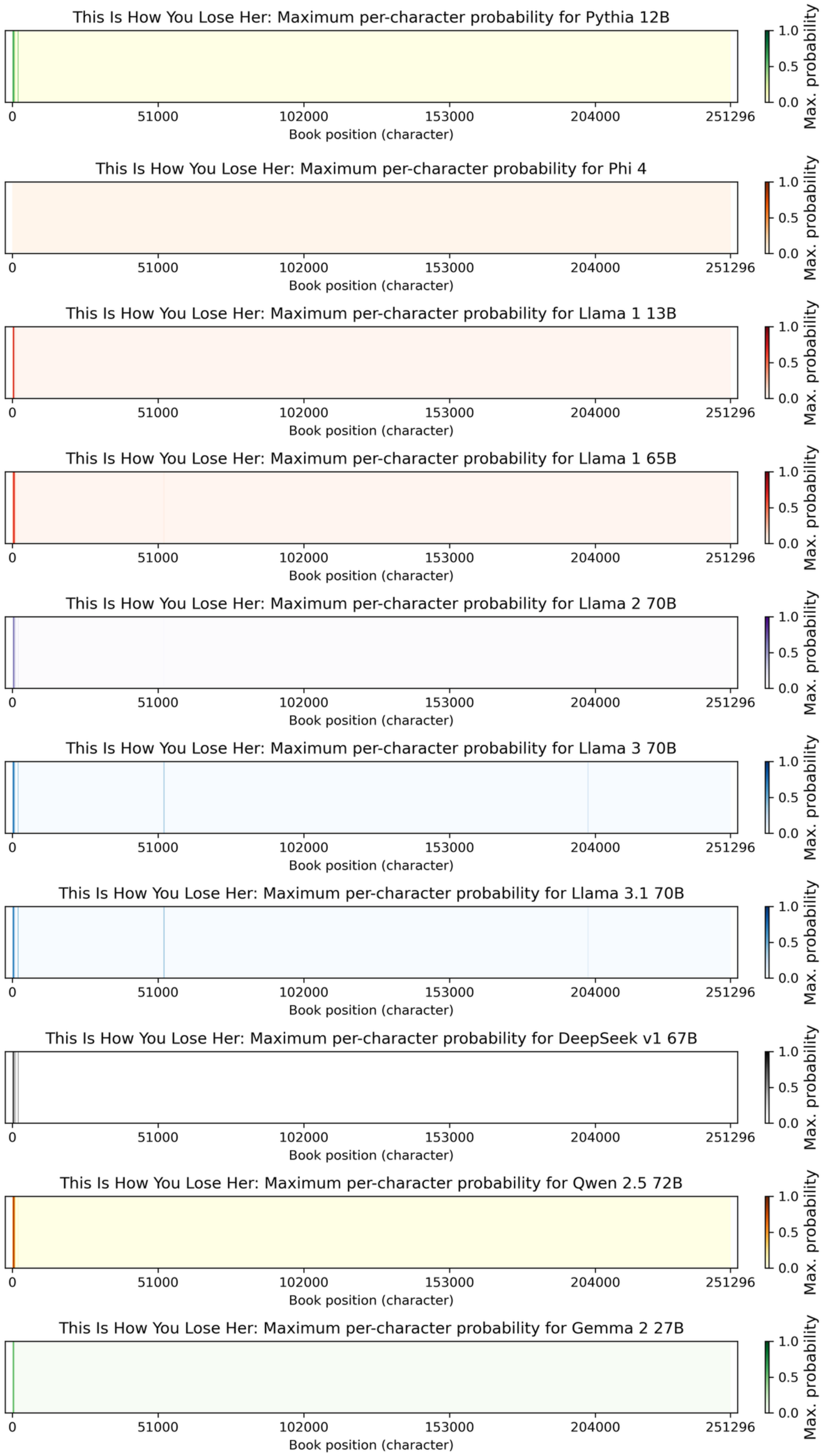
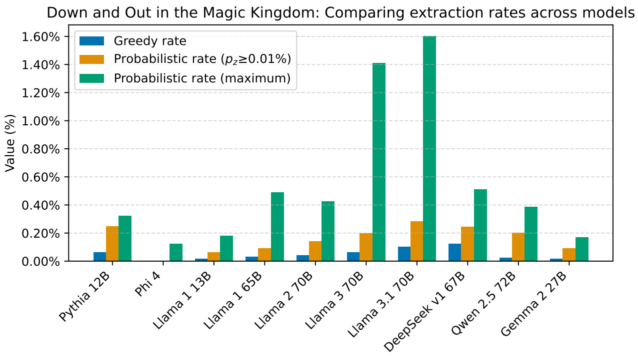


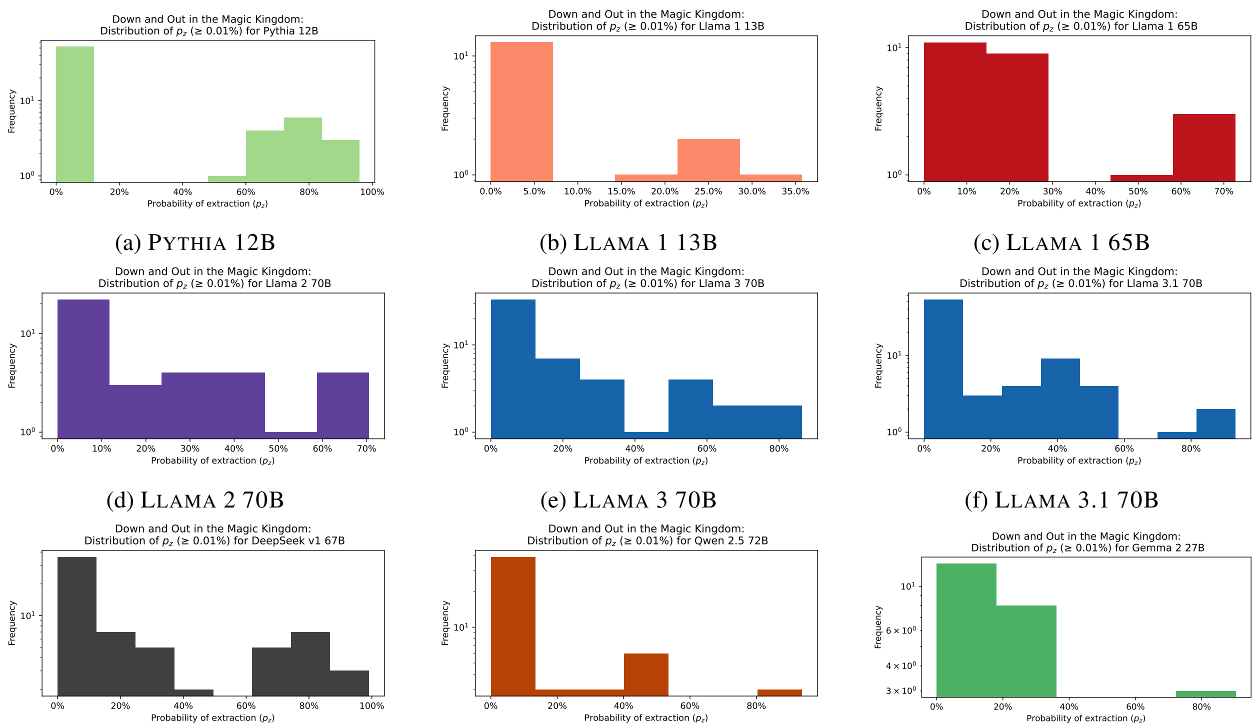
Figure 45: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.13. *Down and Out in the Magic Kingdom*, DOCTOROW



Down and Out in the Magic Kingdom				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.06%	0.25%	0.32%	00:16:40.70
PHI 4	0.00%	0.00%	0.12%	00:35:27.50
LLAMA 1 13B	0.01%	0.06%	0.18%	00:12:37.05
LLAMA 1 65B	0.03%	0.09%	0.49%	00:33:25.06
LLAMA 2 70B	0.04%	0.14%	0.42%	00:29:13.97
LLAMA 3 70B	0.06%	0.20%	1.41%	00:36:03.83
LLAMA 3.1 70B	0.10%	0.28%	1.60%	00:35:05.98
DEEPSEEK V1 67B	0.12%	0.24%	0.51%	00:46:46.42
QWEN 2.5 72B	0.02%	0.20%	0.39%	00:51:43.93
GEMMA 2 27B	0.01%	0.09%	0.17%	00:42:41.89

Figure 46: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



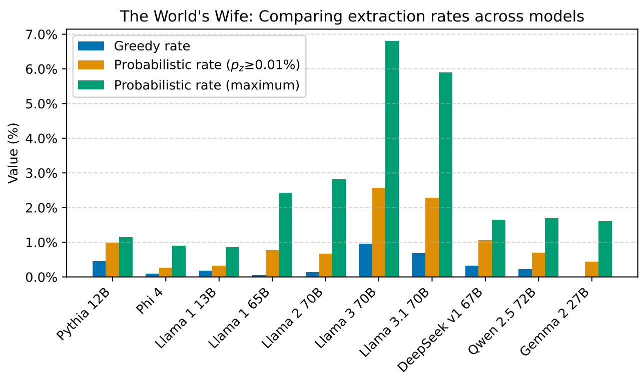
(a) PYTHIA 12B (b) LLAMA 1 13B (c) LLAMA 1 65B

(d) LLAMA 2 70B (e) LLAMA 3 70B (f) LLAMA 3.1 70B

(g) DEEPSEEK V1 67B (h) QWEN 2.5 72B (i) GEMMA 2 27B

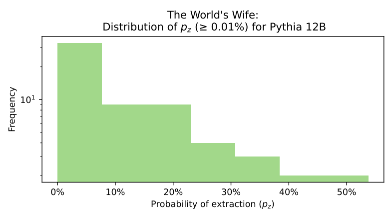
Figure 47: *Down and Out in the Magic Kingdom*, Doctorow (2003): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Only showing models for which there exist any $p_{\boldsymbol{z}} \geq 0.01\%$.

F.1.14. *The World's Wife*, DUFFY

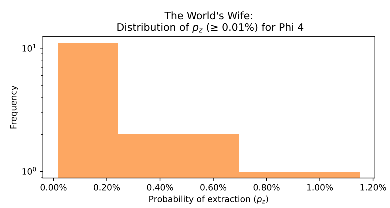


The World's Wife				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.46%	0.98%	1.14%	00:02:16.06
PHI 4	0.10%	0.27%	0.90%	00:04:44.13
LLAMA 1 13B	0.17%	0.32%	0.86%	00:02:01.06
LLAMA 1 65B	0.05%	0.78%	2.43%	00:07:50.11
LLAMA 2 70B	0.14%	0.67%	2.82%	00:07:47.68
LLAMA 3 70B	0.95%	2.57%	6.81%	00:08:56.11
LLAMA 3.1 70B	0.68%	2.28%	5.90%	00:09:29.81
DEEPSEEK V1 67B	0.32%	1.06%	1.65%	00:20:36.99
QWEN 2.5 72B	0.22%	0.70%	1.70%	00:10:05.30
GEMMA 2 27B	0.00%	0.44%	1.60%	00:17:30.78

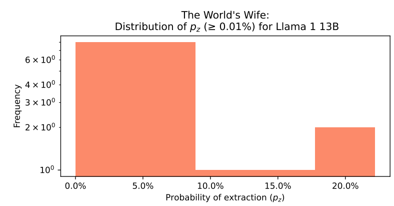
Figure 48: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



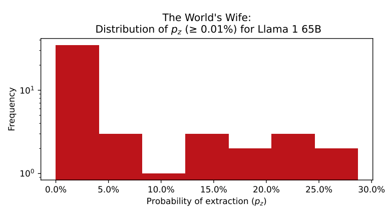
(a) PYTHIA 12B



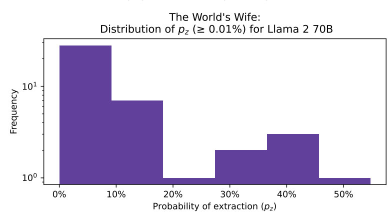
(b) PHI 4 (14B)



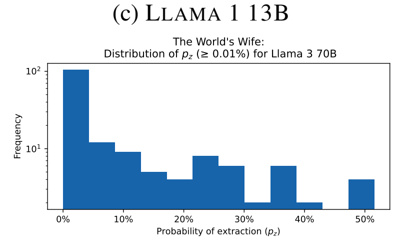
(c) LLAMA 1 13B



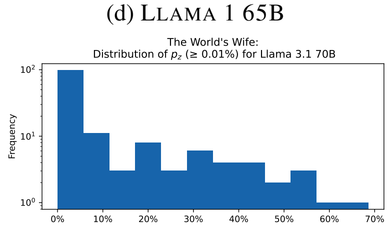
(d) LLAMA 1 65B



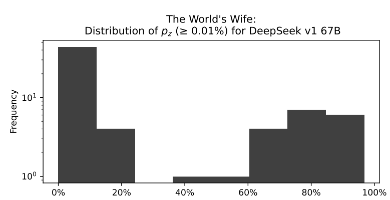
(e) LLAMA 2 70B



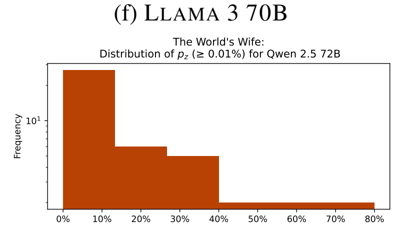
(f) LLAMA 3 70B



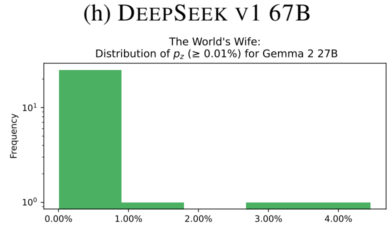
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 49: *The World's Wife*, Duffy (2001): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

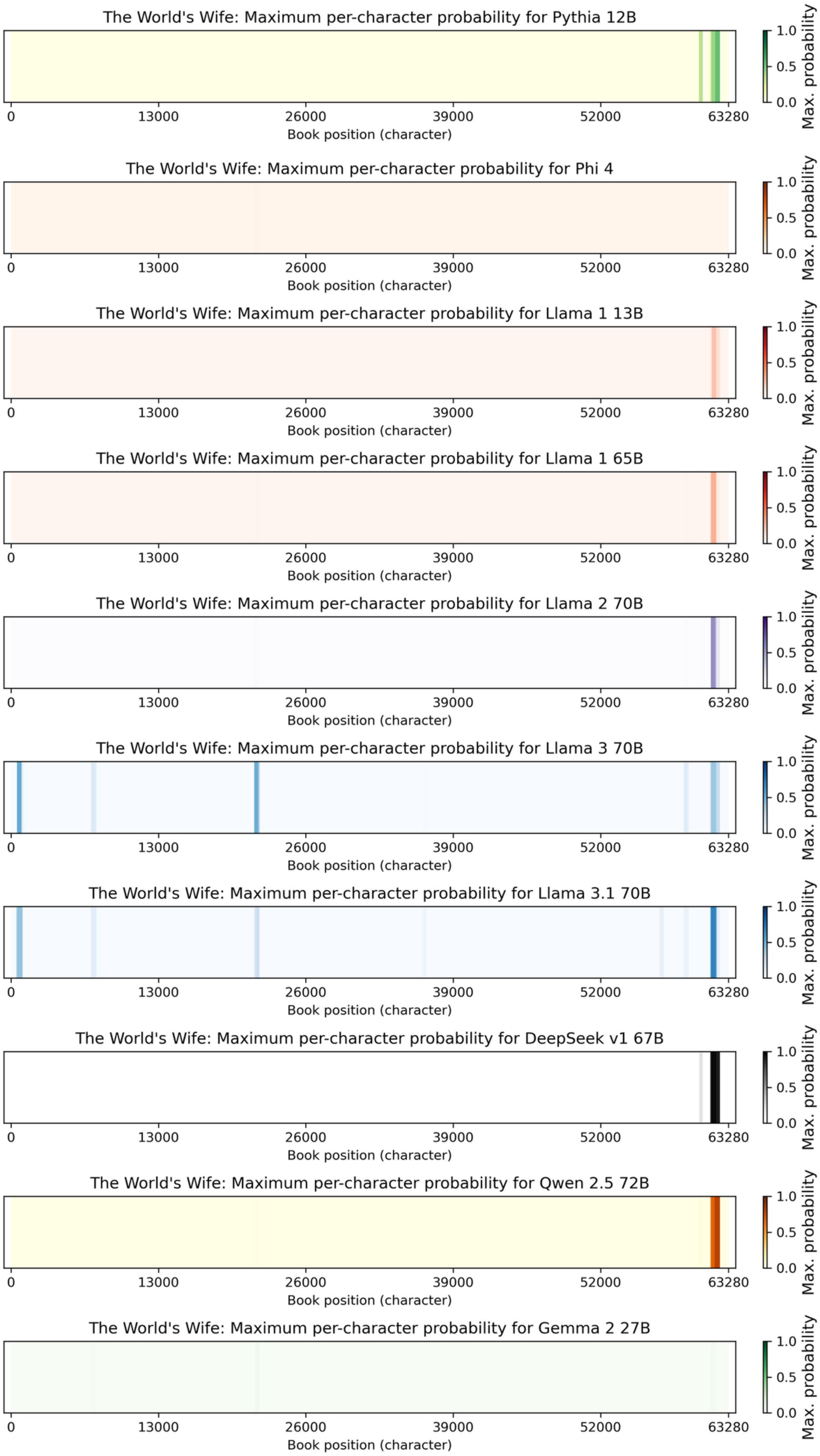
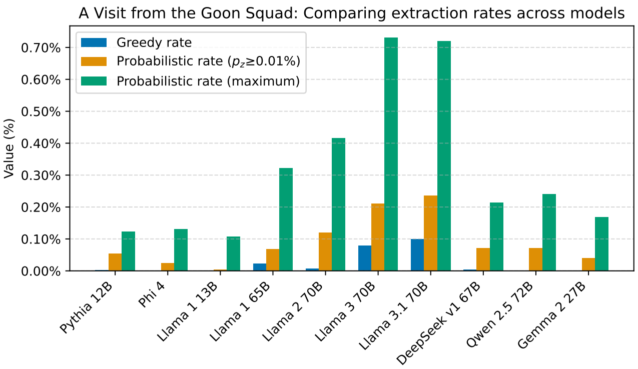


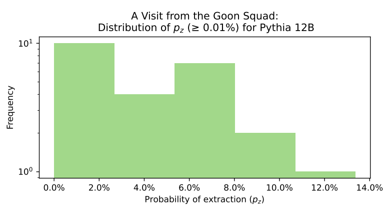
Figure 50: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.15. *A Visit from the Goon Squad*, EGAN

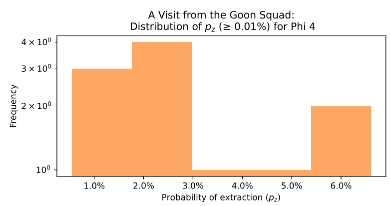


A Visit from the Goon Squad				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.16%	0.48%	0.91%	00:20:05.24
PHI 4	1.00%	3.11%	6.68%	00:36:12.82
LLAMA 1 13B	1.33%	4.60%	21.53%	00:13:07.86
LLAMA 1 65B	8.22%	26.69%	48.99%	00:34:03.40
LLAMA 2 70B	7.26%	26.30%	51.06%	00:29:51.29
LLAMA 3 70B	12.33%	36.81%	64.90%	00:35:47.91
LLAMA 3.1 70B	16.11%	41.60%	67.86%	00:36:24.40
DEEPSEEK V1 67B	2.18%	7.66%	26.94%	00:46:31.71
QWEN 2.5 72B	3.24%	9.89%	29.73%	00:52:35.89
GEMMA 2 27B	2.34%	9.71%	31.88%	00:44:44.88

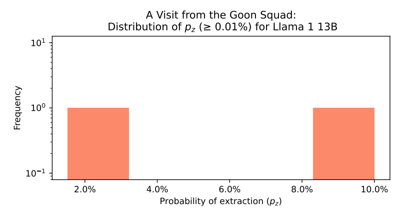
Figure 51: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



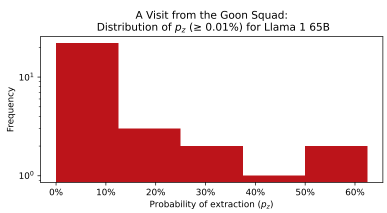
(a) PYTHIA 12B



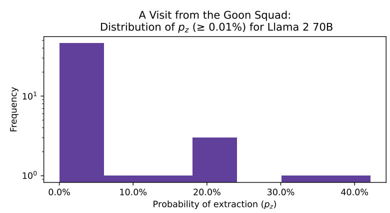
(b) PHI 4 (14B)



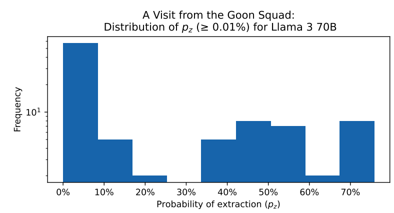
(c) LLAMA 1 13B



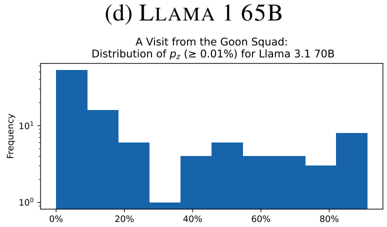
(d) LLAMA 1 65B



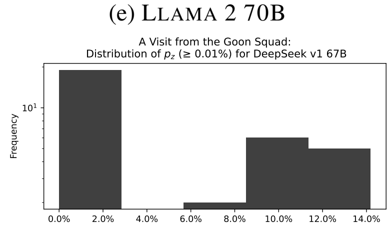
(e) LLAMA 2 70B



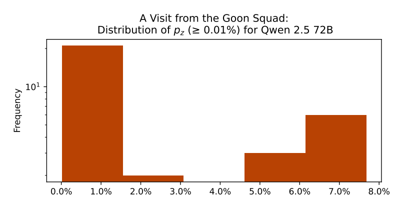
(f) LLAMA 3 70B



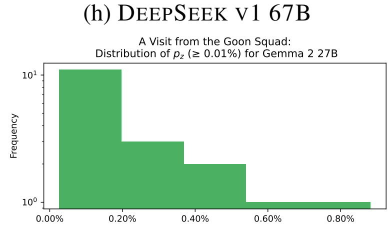
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 52: *A Visit from the Goon Squad*, Egan (2010): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

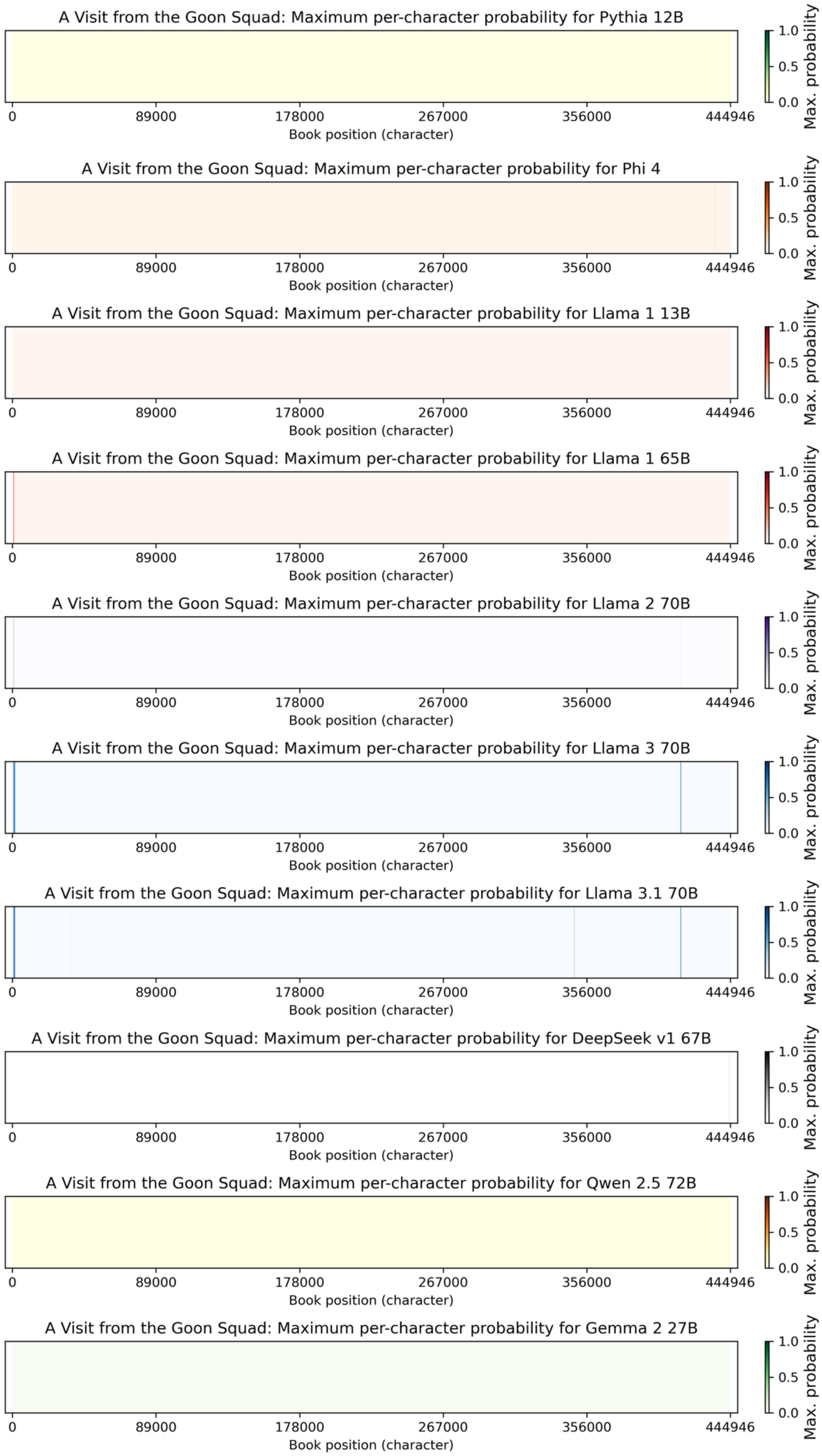
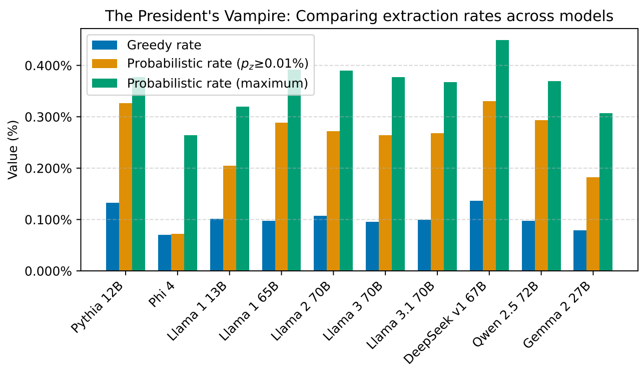


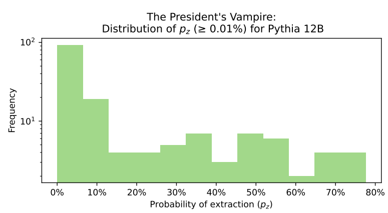
Figure 53: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.16. *The President's Vampire*, FARNSWORTH

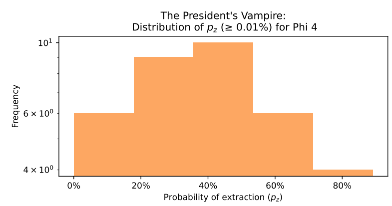


The President's Vampire				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.02%	0.10%	00:15:06.06
PHI 4	0.05%	0.10%	0.35%	00:28:41.59
LLAMA 1 13B	0.00%	0.06%	0.35%	00:16:46.39
LLAMA 1 65B	0.09%	0.55%	1.84%	00:58:18.24
LLAMA 2 70B	0.27%	1.06%	2.60%	00:57:14.62
LLAMA 3 70B	1.06%	2.95%	8.26%	01:09:13.78
LLAMA 3.1 70B	1.52%	3.93%	9.75%	01:10:59.00
DEEPSEEK V1 67B	0.02%	0.08%	0.49%	01:16:38.75
QWEN 2.5 72B	0.05%	0.36%	1.23%	01:18:30.16
GEMMA 2 27B	0.01%	0.16%	0.73%	01:07:15.42

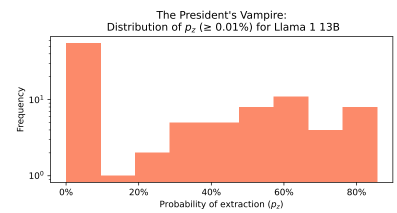
Figure 54: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



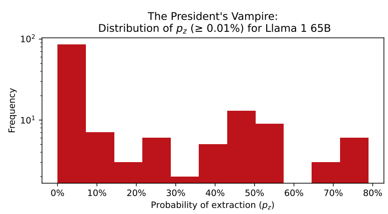
(a) PYTHIA 12B



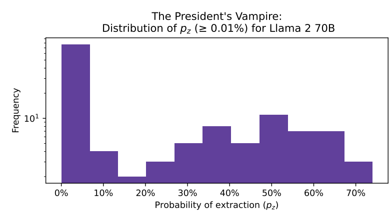
(b) PHI 4 (14B)



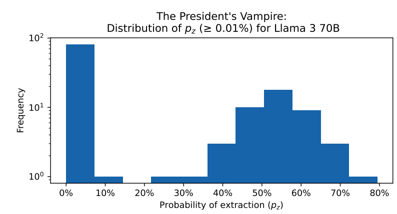
(c) LLAMA 1 13B



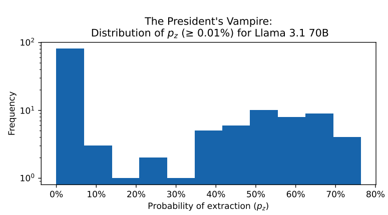
(d) LLAMA 1 65B



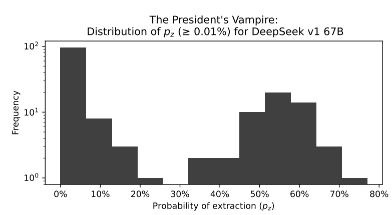
(e) LLAMA 2 70B



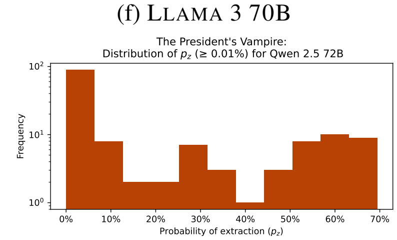
(f) LLAMA 3 70B



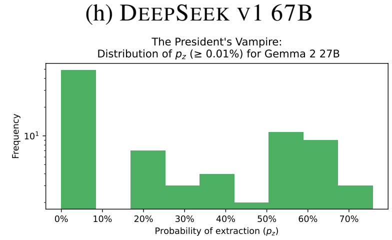
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



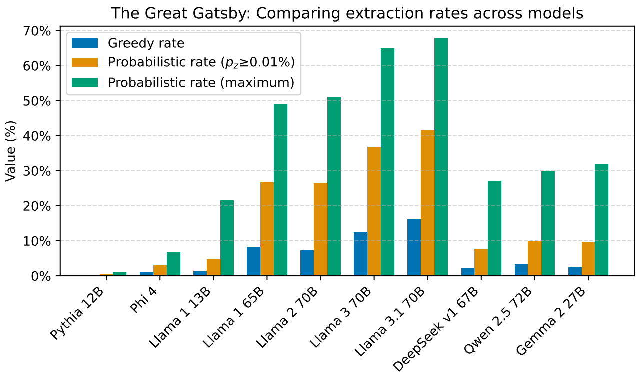
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

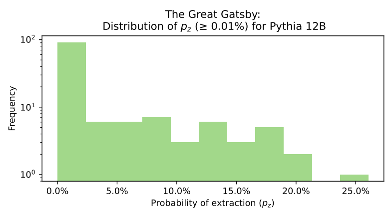
Figure 55: *The President's Vampire*, Farnsworth (2011): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.17. *The Great Gatsby*, FITZGERALD

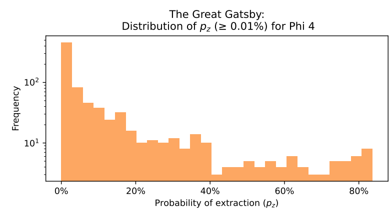


The Great Gatsby				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.16%	0.48%	0.91%	00:20:05.24
PHI 4	1.00%	3.11%	6.68%	00:36:12.82
LLAMA 1 13B	1.33%	4.60%	21.53%	00:13:07.86
LLAMA 1 65B	8.22%	26.69%	48.99%	00:34:03.40
LLAMA 2 70B	7.26%	26.30%	51.06%	00:29:51.29
LLAMA 3 70B	12.33%	36.81%	64.90%	00:35:47.91
LLAMA 3.1 70B	16.11%	41.60%	67.86%	00:36:24.40
DEEPSEEK V1 67B	2.18%	7.66%	26.94%	00:46:31.71
QWEN 2.5 72B	3.24%	9.89%	29.73%	00:52:35.89
GEMMA 2 27B	2.34%	9.71%	31.88%	00:44:44.88

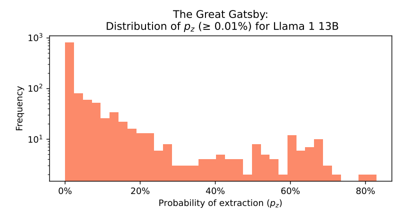
Figure 56: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



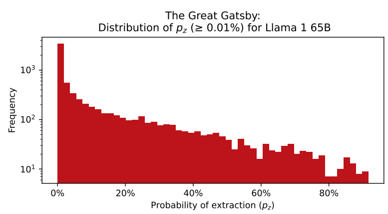
(a) PYTHIA 12B



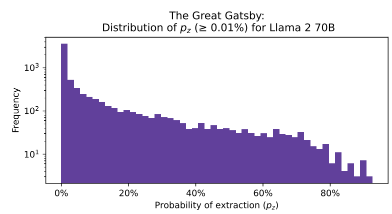
(b) PHI 4 (14B)



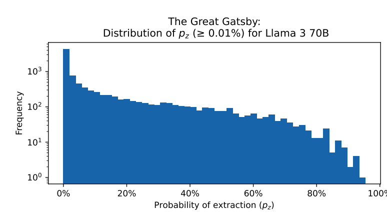
(c) LLAMA 1 13B



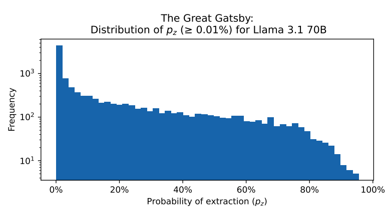
(d) LLAMA 1 65B



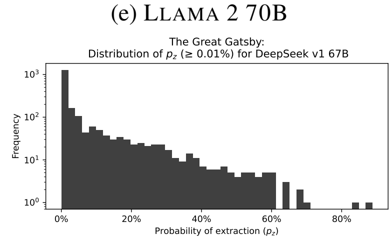
(e) LLAMA 2 70B



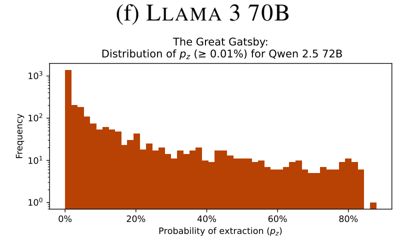
(f) LLAMA 3 70B



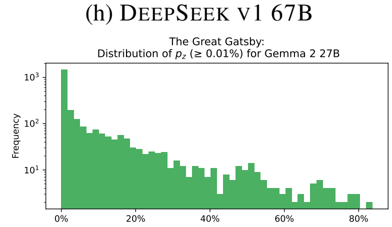
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 57: *The Great Gatsby*, Fitzgerald (1925): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models.

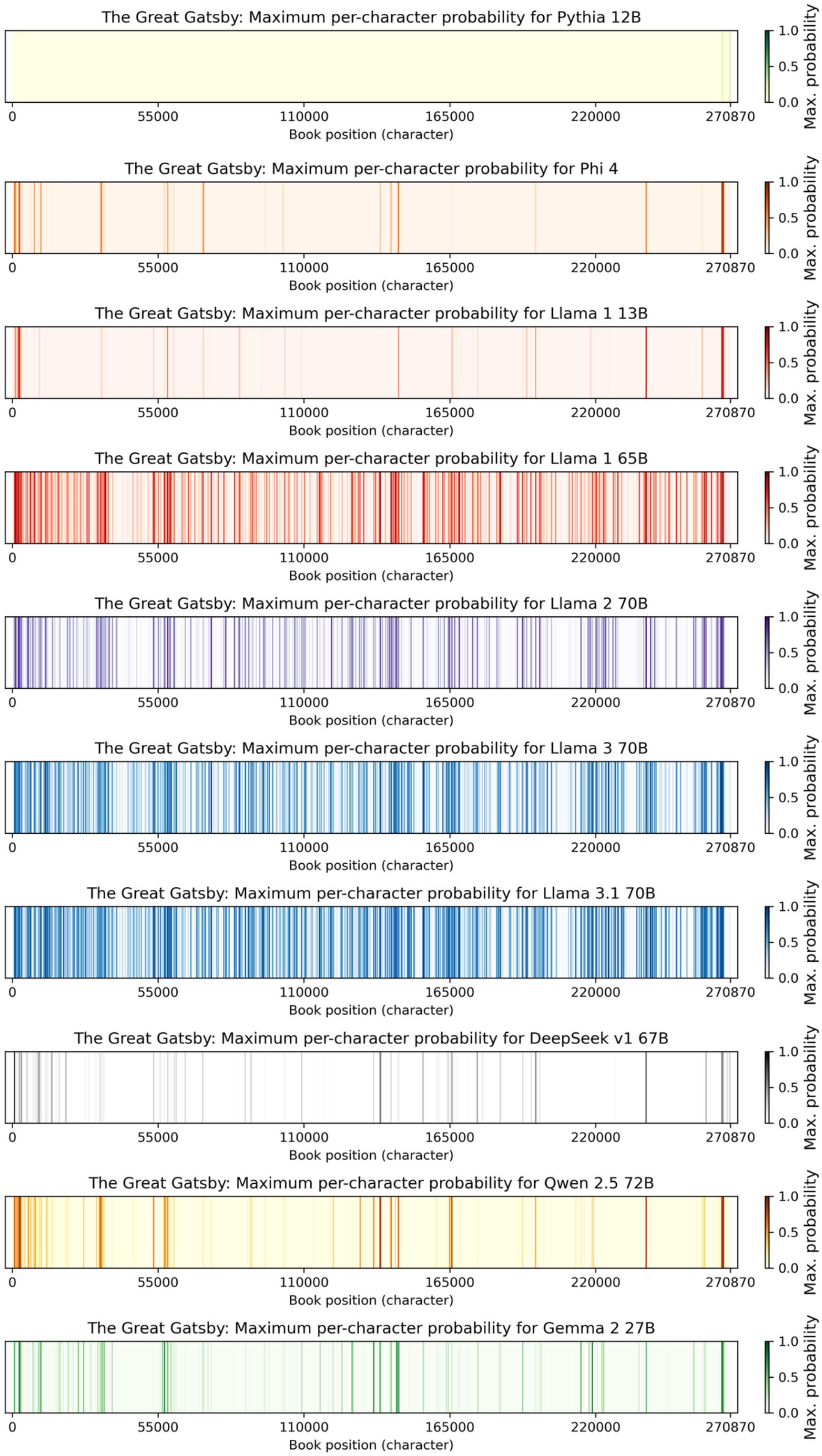
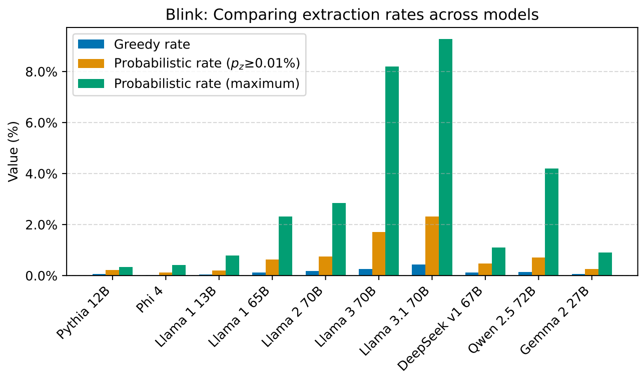


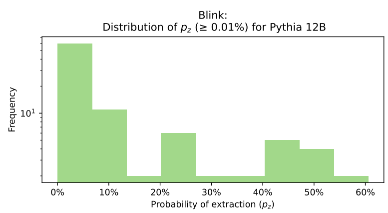
Figure 58: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.18. *Blink*, GLADWELL

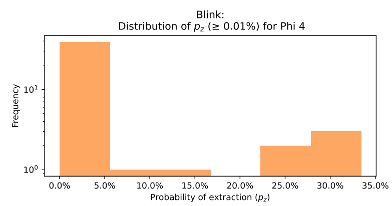


Blink				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.05%	0.21%	0.33%	00:14:29.08
PHI 4	0.01%	0.10%	0.41%	00:24:16.36
LLAMA 1 13B	0.03%	0.19%	0.77%	00:13:33.43
LLAMA 1 65B	0.12%	0.62%	2.32%	00:48:36.54
LLAMA 2 70B	0.17%	0.73%	2.83%	00:48:40.22
LLAMA 3 70B	0.25%	1.71%	8.20%	00:59:12.29
LLAMA 3.1 70B	0.43%	2.31%	9.26%	00:59:11.72
DEEPSEEK V1 67B	0.11%	0.46%	1.10%	01:08:40.10
QWEN 2.5 72B	0.13%	0.70%	4.19%	01:02:51.21
GEMMA 2 27B	0.04%	0.25%	0.89%	01:03:43.75

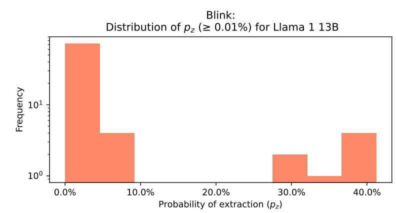
Figure 59: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n,p)-discoverable rate across different models.



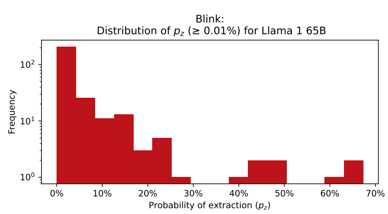
(a) PYTHIA 12B



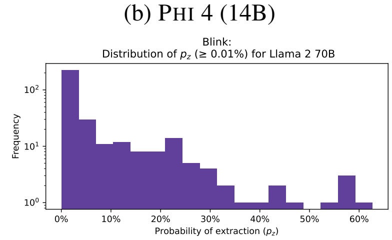
(b) PHI 4 (14B)



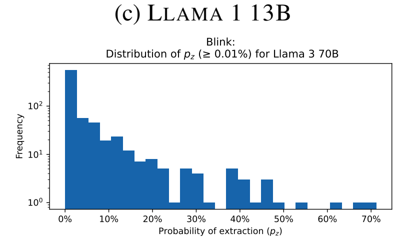
(c) LLAMA 1 13B



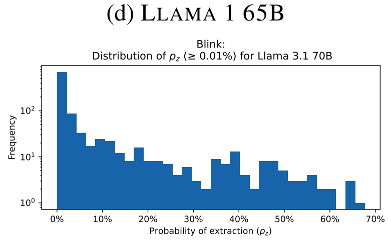
(d) LLAMA 1 65B



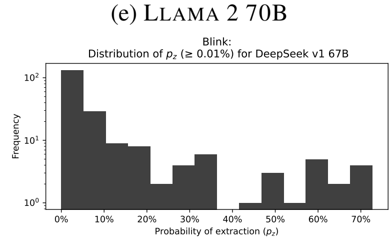
(e) LLAMA 2 70B



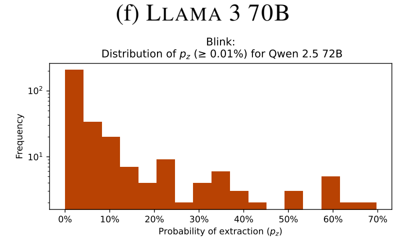
(f) LLAMA 3 70B



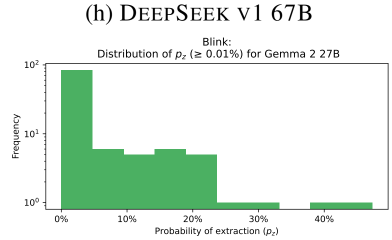
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 60: *Blink*, Gladwell (2005): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

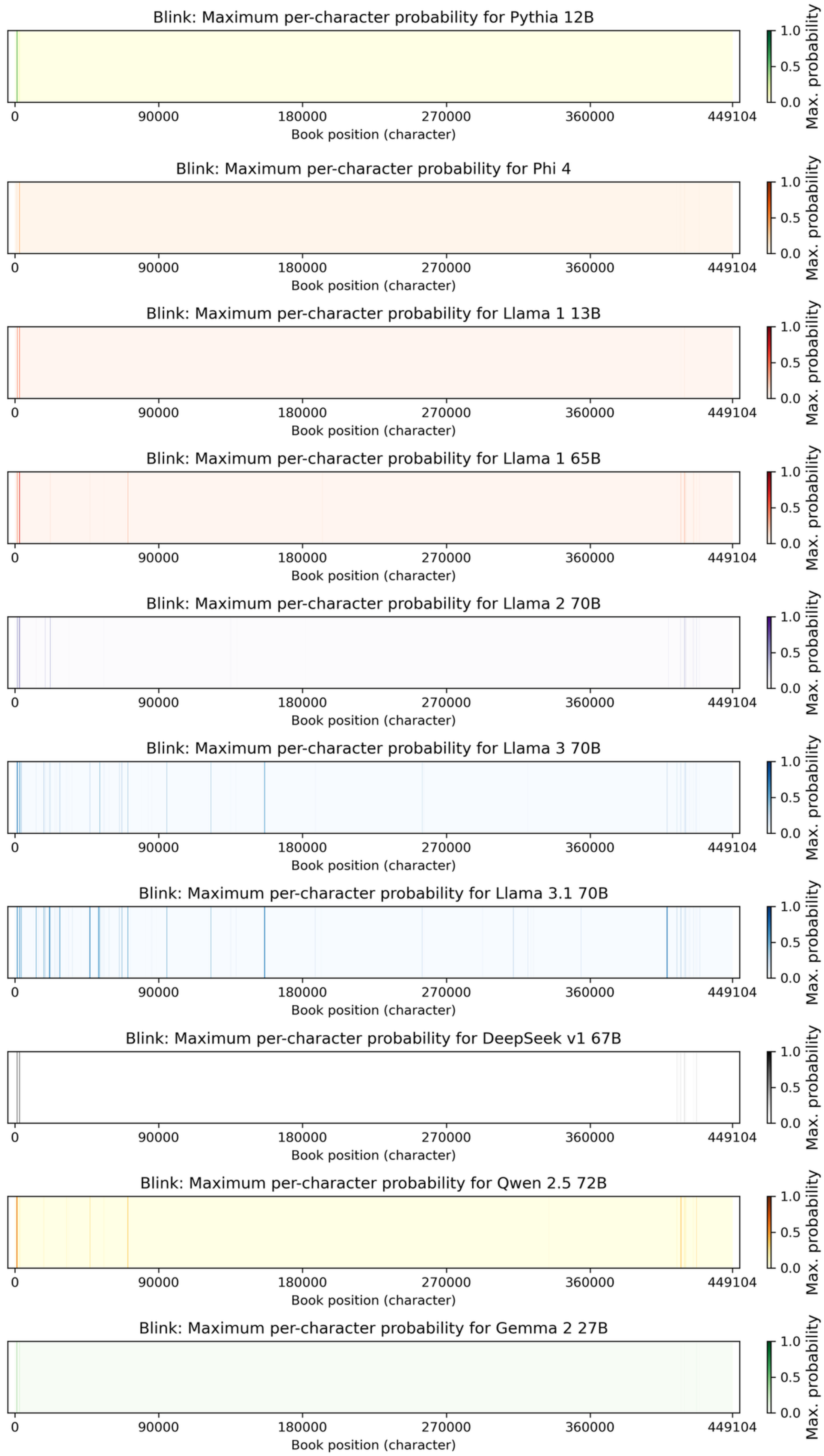
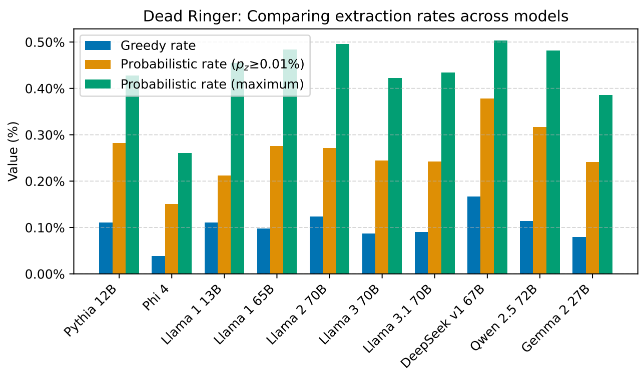


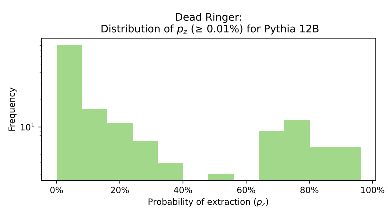
Figure 61: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.19. *Dead Ringer*, GOLDEN

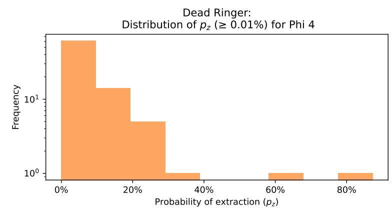


Dead Ringer				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.11%	0.28%	0.43%	00:16:42.57
PHI 4	0.04%	0.15%	0.26%	00:30:56.08
LLAMA 1 13B	0.11%	0.21%	0.45%	00:17:39.28
LLAMA 1 65B	0.10%	0.28%	0.48%	00:58:41.88
LLAMA 2 70B	0.12%	0.27%	0.50%	00:59:13.07
LLAMA 3 70B	0.09%	0.24%	0.42%	01:12:25.79
LLAMA 3.1 70B	0.09%	0.24%	0.43%	01:11:09.37
DEEPSEEK V1 67B	0.17%	0.38%	0.50%	01:20:36.57
QWEN 2.5 72B	0.11%	0.32%	0.48%	01:30:30.56
GEMMA 2 27B	0.08%	0.24%	0.39%	01:20:40.55

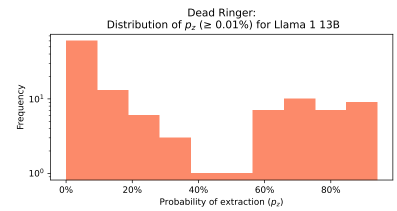
Figure 62: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



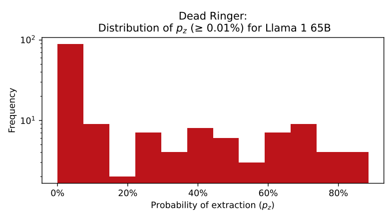
(a) PYTHIA 12B



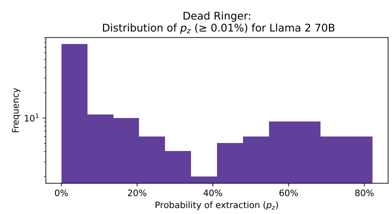
(b) PHI 4 (14B)



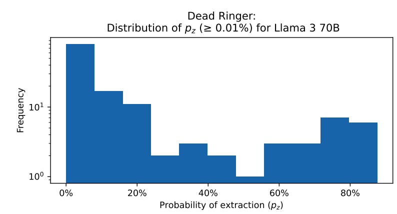
(c) LLAMA 1 13B



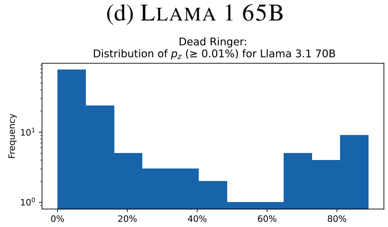
(d) LLAMA 1 65B



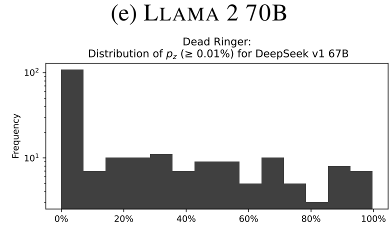
(e) LLAMA 2 70B



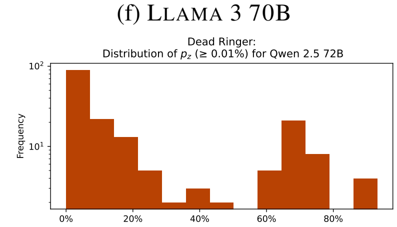
(f) LLAMA 3 70B



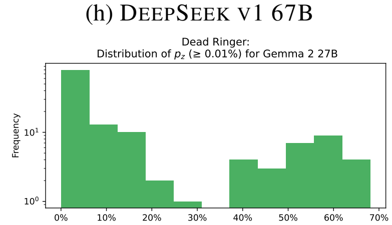
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



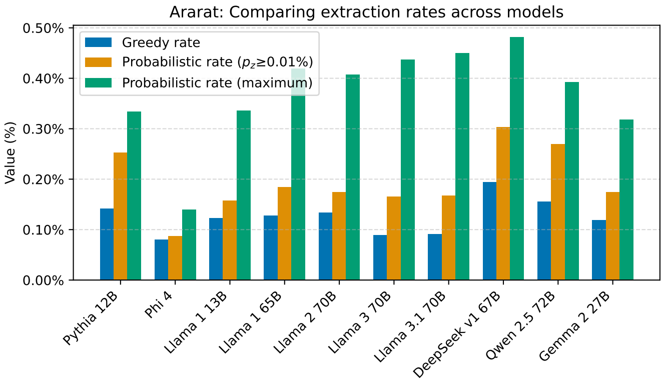
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

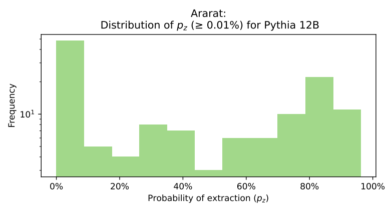
Figure 63: *Dead Ringer*, Golden (2016): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models.

F.1.20. *Ararat*, GOLDEN

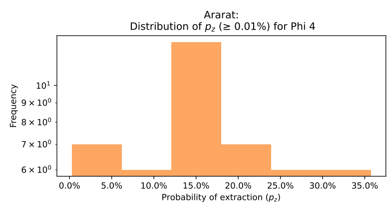


	Ararat			
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.14%	0.25%	0.33%	00:30:36.62
PHI 4	0.08%	0.09%	0.14%	01:02:21.74
LLAMA 1 13B	0.12%	0.16%	0.34%	00:22:18.90
LLAMA 1 65B	0.13%	0.18%	0.42%	01:02:33.58
LLAMA 2 70B	0.13%	0.17%	0.41%	00:55:30.87
LLAMA 3 70B	0.09%	0.16%	0.44%	01:08:08.25
LLAMA 3.1 70B	0.09%	0.17%	0.45%	01:07:24.99
DEEPSEEK V1 67B	0.19%	0.30%	0.48%	01:07:58.66
QWEN 2.5 72B	0.16%	0.27%	0.39%	01:18:27.52
GEMMA 2 27B	0.12%	0.17%	0.32%	01:11:25.61

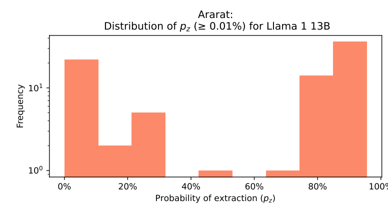
Figure 64: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



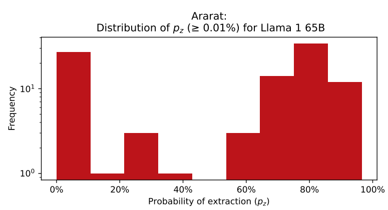
(a) PYTHIA 12B



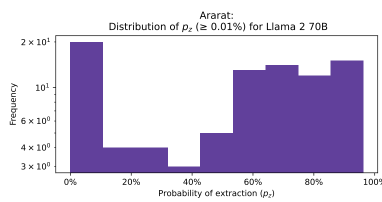
(b) PHI 4 (14B)



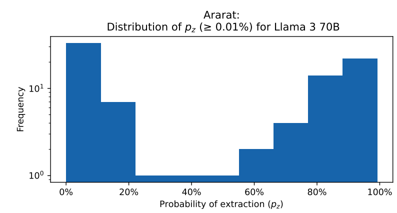
(c) LLAMA 1 13B



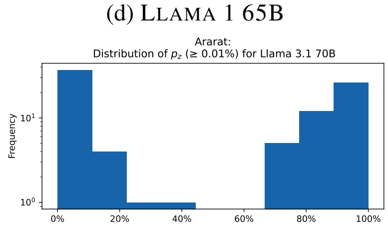
(d) LLAMA 1 65B



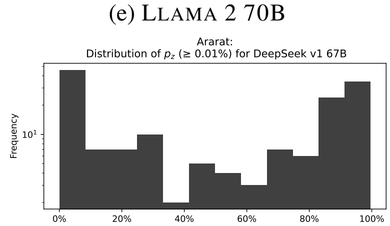
(e) LLAMA 2 70B



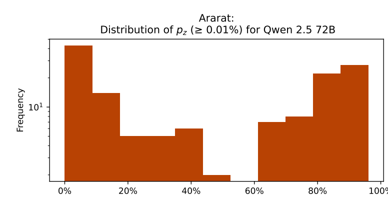
(f) LLAMA 3 70B



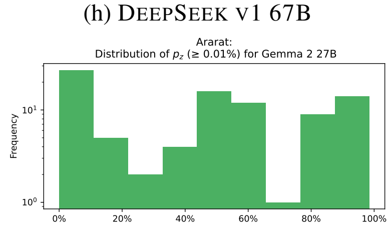
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



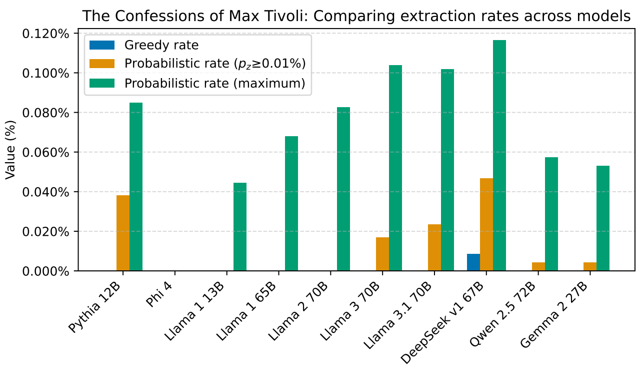
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

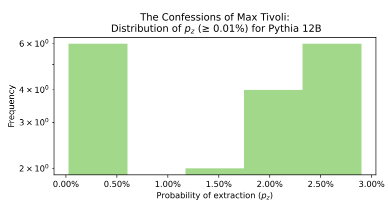
Figure 65: *Ararat*, Golden (2017): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.21. *The Confessions of Max Tivoli*, GREER

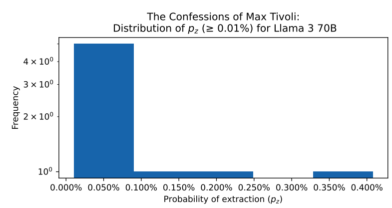


The Confessions of Max Tivoli				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.04%	0.08%	00:31:23.50
PHI 4	0.00%	0.00%	0.00%	01:03:00.43
LLAMA 1 13B	0.00%	0.00%	0.04%	00:22:51.88
LLAMA 1 65B	0.00%	0.00%	0.07%	00:59:47.86
LLAMA 2 70B	0.00%	0.00%	0.08%	00:51:14.57
LLAMA 3 70B	0.00%	0.02%	0.10%	01:01:40.94
LLAMA 3.1 70B	0.00%	0.02%	0.10%	01:03:07.68
DEEPSEEK V1 67B	0.01%	0.05%	0.12%	01:09:46.77
QWEN 2.5 72B	0.00%	0.00%	0.06%	01:21:13.67
GEMMA 2 27B	0.00%	0.00%	0.05%	01:09:55.07

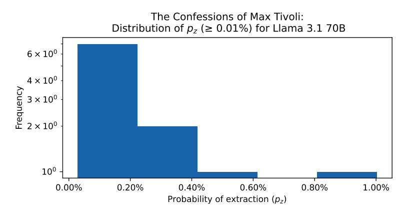
Figure 66: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



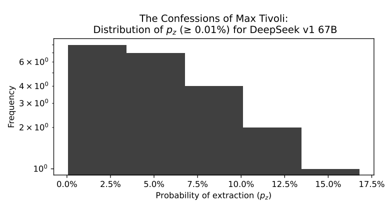
(a) PYTHIA 12B



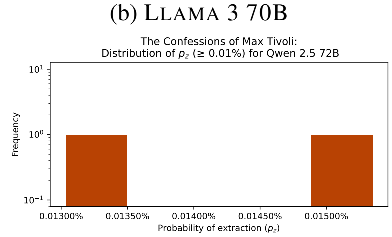
(b) LLAMA 3 70B



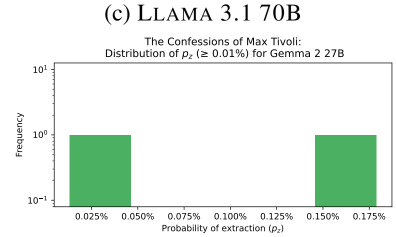
(c) LLAMA 3.1 70B



(d) DEEPSEEK V1 67B



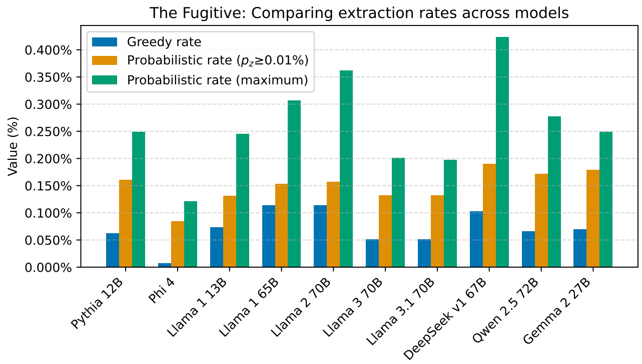
(e) QWEN 2.5 72B



(f) GEMMA 2 27B

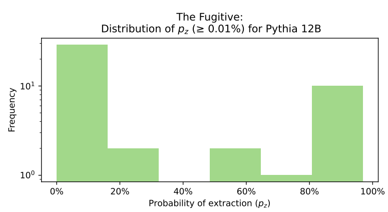
Figure 67: *The Confessions of Max Tivoli*, Greer (2005): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models. Only showing models for which there exist any $p_z \geq 0.01\%$.

F.1.22. *Theodore Boone: The Fugitive*, GRISHAM

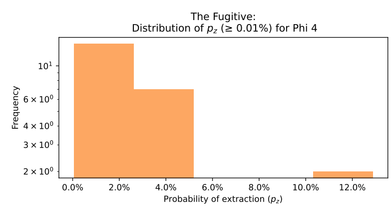


The Fugitive				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.06%	0.16%	0.25%	00:08:14.74
PHI 4	0.01%	0.08%	0.12%	00:15:42.06
LLAMA 1 13B	0.07%	0.13%	0.24%	00:08:12.61
LLAMA 1 65B	0.11%	0.15%	0.31%	00:30:03.36
LLAMA 2 70B	0.11%	0.16%	0.36%	00:30:37.72
LLAMA 3 70B	0.05%	0.13%	0.20%	00:37:01.28
LLAMA 3.1 70B	0.05%	0.13%	0.20%	00:37:03.15
DEEPSEEK V1 67B	0.10%	0.19%	0.42%	00:46:17.99
QWEN 2.5 72B	0.07%	0.17%	0.28%	00:40:28.49
GEMMA 2 27B	0.07%	0.18%	0.25%	00:41:54.08

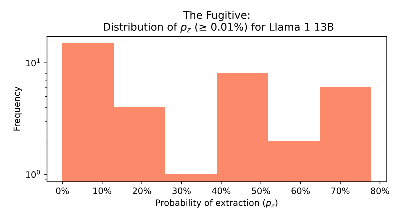
Figure 68: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



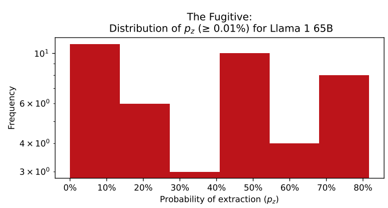
(a) PYTHIA 12B



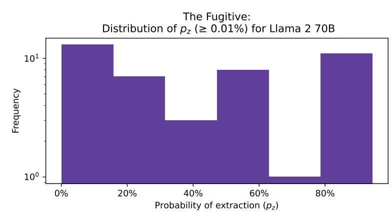
(b) PHI 4 (14B)



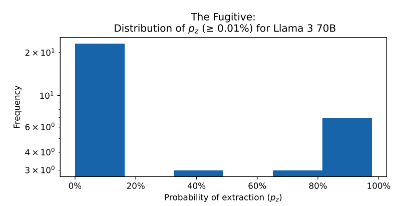
(c) LLAMA 1 13B



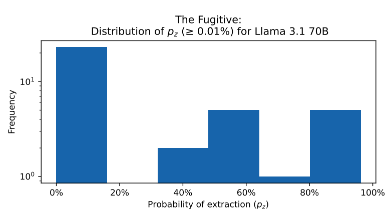
(d) LLAMA 1 65B



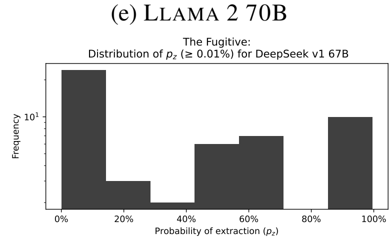
(e) LLAMA 2 70B



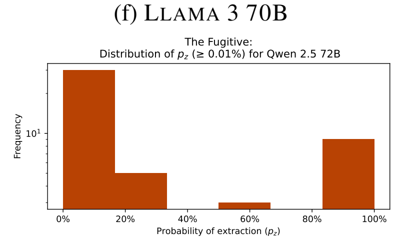
(f) LLAMA 3 70B



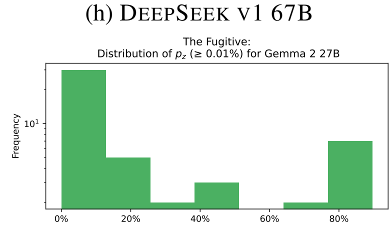
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



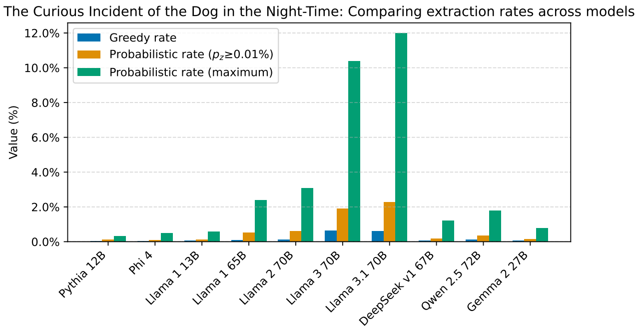
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

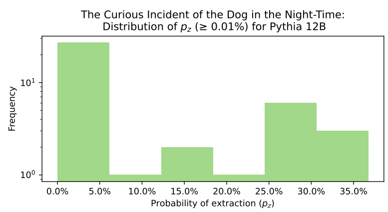
Figure 69: *Theodore Boone: The Fugitive*, Grisham (2015): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.23. *The Curious Incident of the Dog in the Night-Time*, HADDON

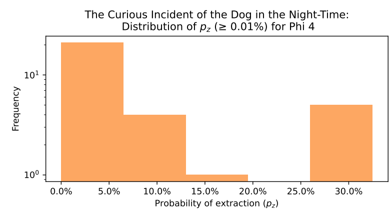


The Curious Incident of the Dog in the Night-Time				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.03%	0.12%	0.32%	00:10:16.88
PHI 4	0.03%	0.10%	0.49%	00:18:42.60
LLAMA 1 13B	0.06%	0.12%	0.59%	00:10:00.99
LLAMA 1 65B	0.10%	0.52%	2.39%	00:35:52.83
LLAMA 2 70B	0.13%	0.62%	3.08%	00:36:36.43
LLAMA 3 70B	0.63%	1.92%	10.39%	00:43:59.67
LLAMA 3.1 70B	0.60%	2.29%	11.98%	00:44:27.83
DEEPSEEK V1 67B	0.05%	0.18%	1.20%	00:53:18.28
QWEN 2.5 72B	0.13%	0.34%	1.80%	00:48:15.93
GEMMA 2 27B	0.06%	0.15%	0.79%	00:53:41.68

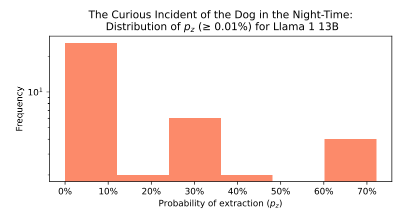
Figure 70: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\mathbf{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



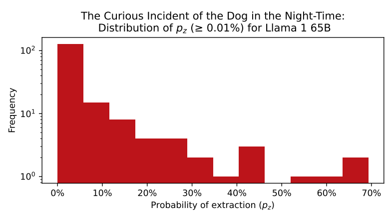
(a) PYTHIA 12B



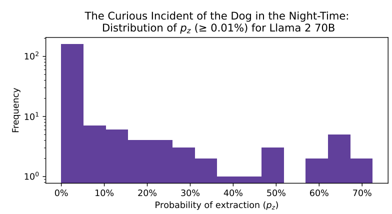
(b) PHI 4 (14B)



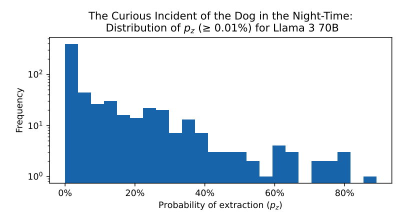
(c) LLAMA 1 13B



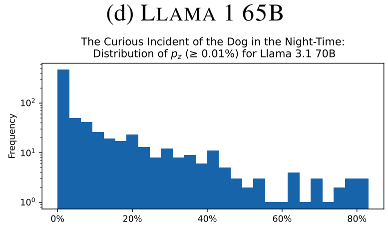
(d) LLAMA 1 65B



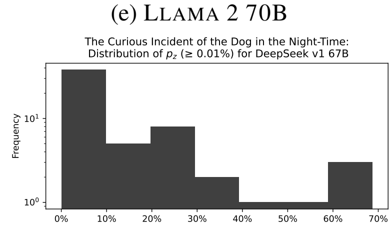
(e) LLAMA 2 70B



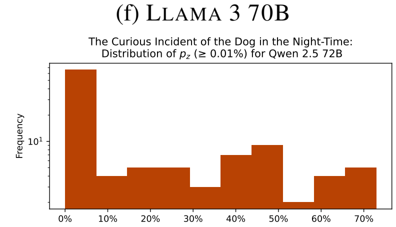
(f) LLAMA 3 70B



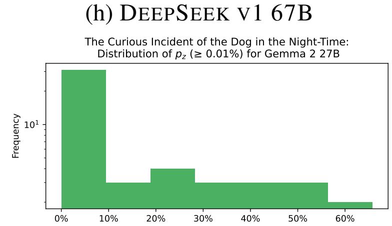
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 71: *The Curious Incident of the Dog in the Night-Time*, Haddon (2003): Distributions over $p_{\mathbf{z}}$ (limited to $p_{\mathbf{z}} \geq 0.01\%$) for different models.

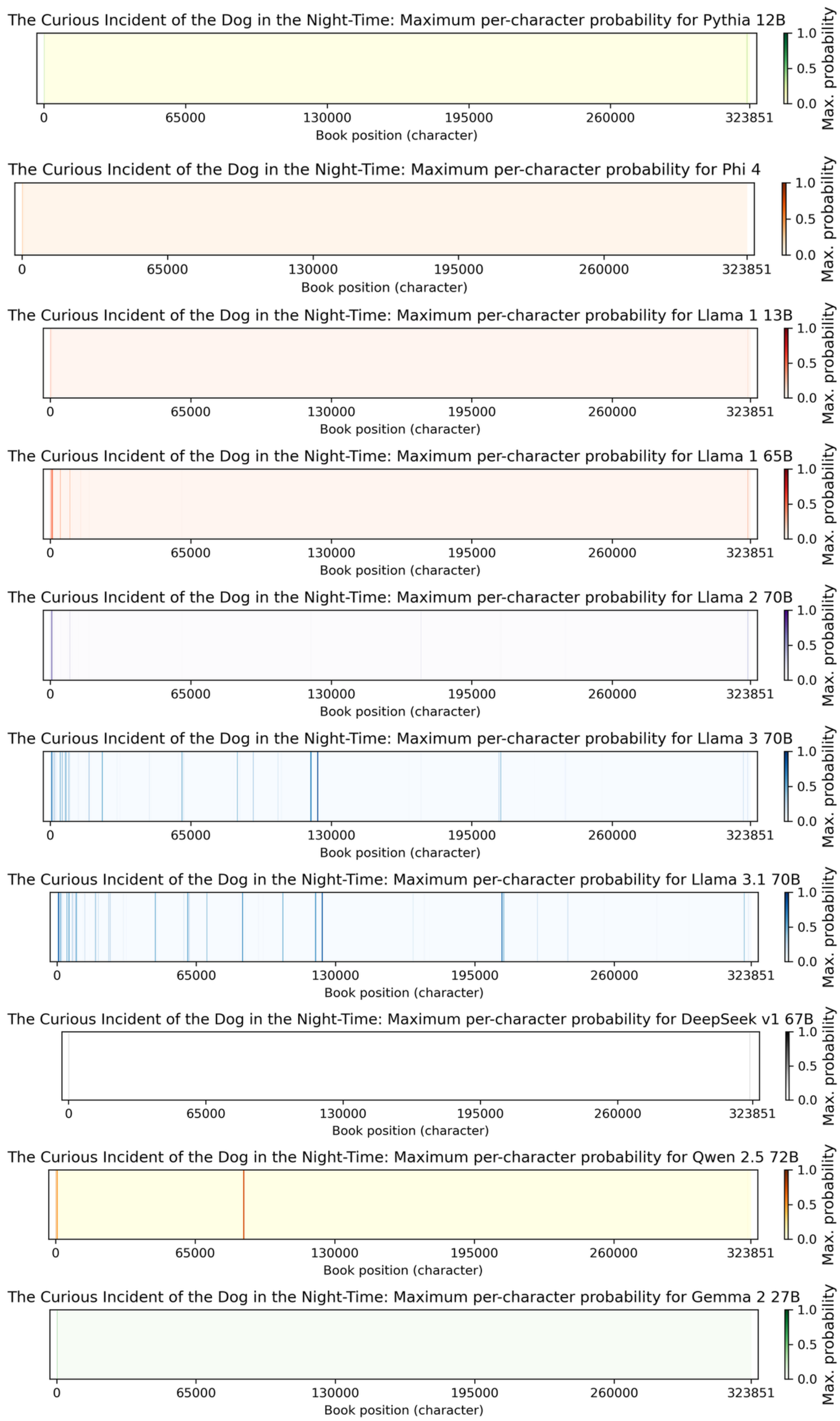
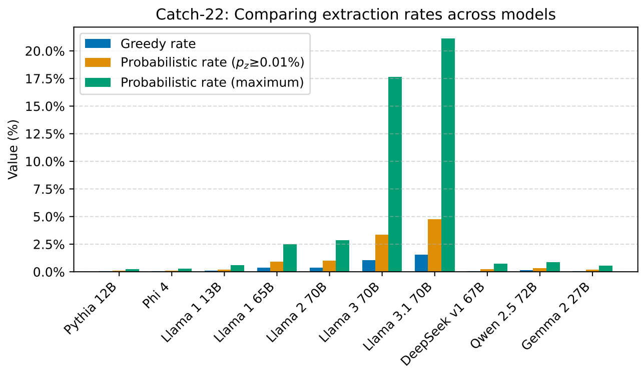


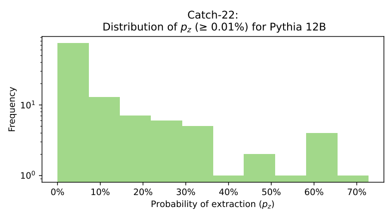
Figure 72: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.24. *Catch-22*, HELLER

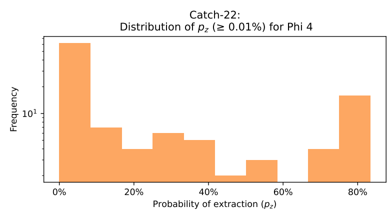


Catch-22				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.03%	0.10%	0.23%	00:34:24.57
PHI 4	0.03%	0.10%	0.26%	00:59:47.42
LLAMA 1 13B	0.07%	0.20%	0.61%	00:34:35.46
LLAMA 1 65B	0.38%	0.89%	2.50%	02:03:04.55
LLAMA 2 70B	0.36%	1.00%	2.87%	02:02:26.37
LLAMA 3 70B	1.03%	3.34%	17.62%	02:37:38.47
LLAMA 3.1 70B	1.56%	4.76%	21.10%	02:28:55.27
DEEPSEEK V1 67B	0.06%	0.22%	0.74%	02:30:53.45
QWEN 2.5 72B	0.13%	0.32%	0.87%	02:37:55.18
GEMMA 2 27B	0.06%	0.18%	0.53%	02:29:00.40

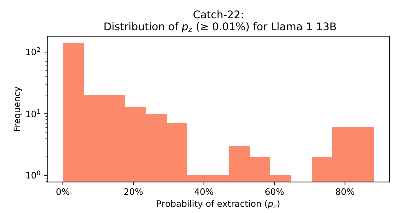
Figure 73: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



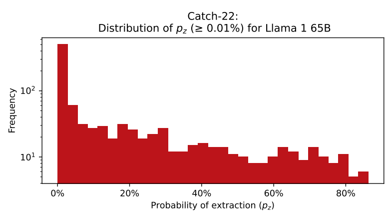
(a) PYTHIA 12B



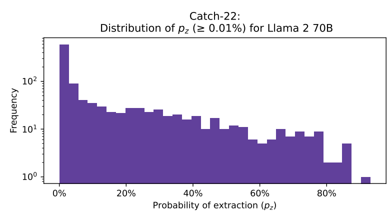
(b) PHI 4 (14B)



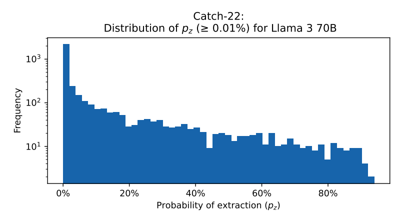
(c) LLAMA 1 13B



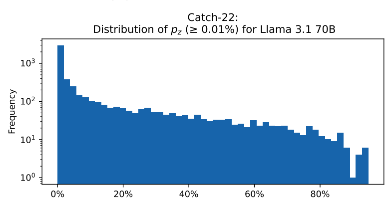
(d) LLAMA 1 65B



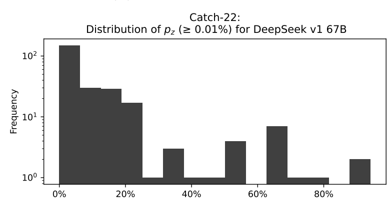
(e) LLAMA 2 70B



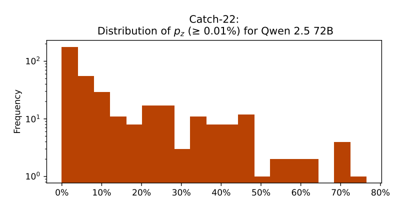
(f) LLAMA 3 70B



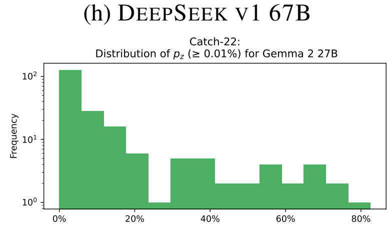
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 74: *Catch-22*, Heller (1961): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

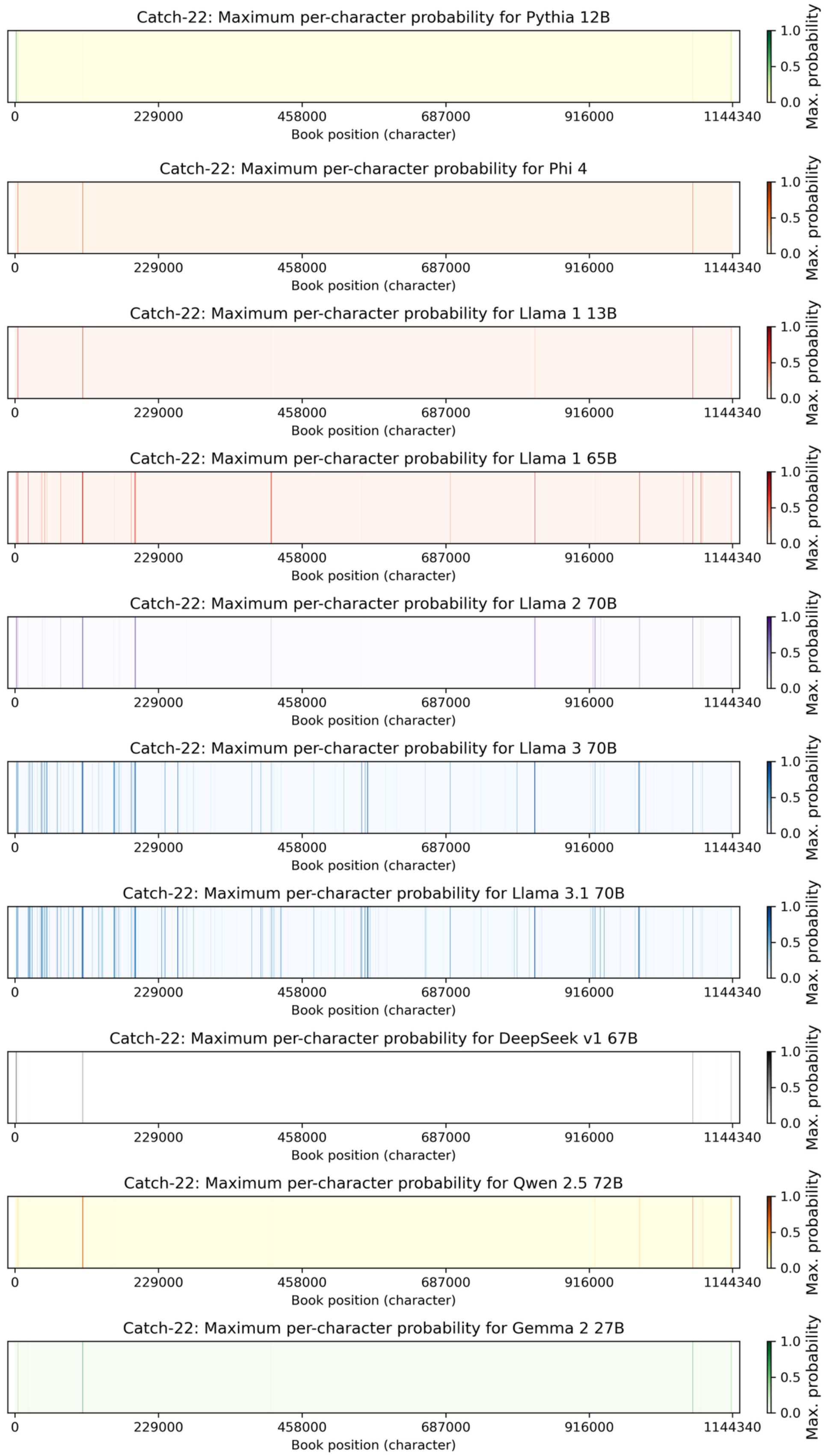
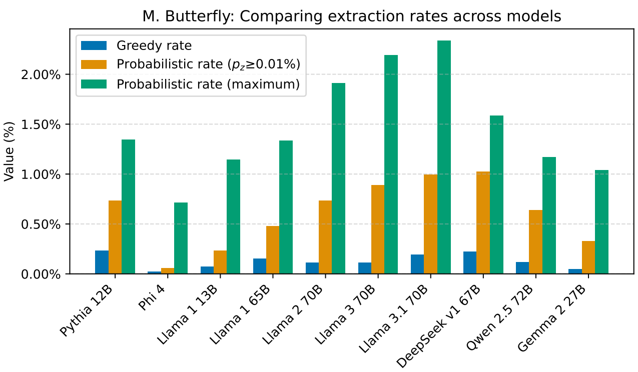


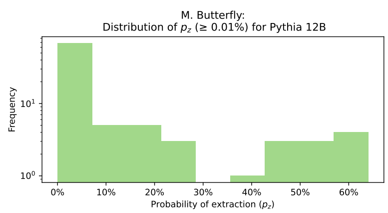
Figure 75: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.25. *M. Butterfly*, HWANG

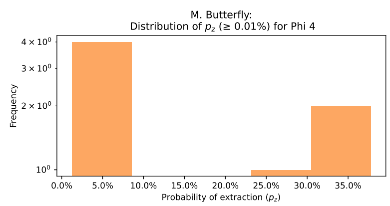


M. Butterfly				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.23%	0.74%	1.34%	00:08:34.80
PHI 4	0.02%	0.06%	0.71%	00:18:57.56
LLAMA 1 13B	0.07%	0.23%	1.14%	00:06:16.90
LLAMA 1 65B	0.15%	0.48%	1.34%	00:16:33.55
LLAMA 2 70B	0.11%	0.74%	1.91%	00:14:04.98
LLAMA 3 70B	0.11%	0.89%	2.19%	00:17:10.41
LLAMA 3.1 70B	0.19%	0.99%	2.34%	00:16:38.24
DEEPSEEK V1 67B	0.22%	1.02%	1.58%	00:27:01.18
QWEN 2.5 72B	0.12%	0.64%	1.17%	00:31:30.45
GEMMA 2 27B	0.05%	0.33%	1.04%	00:25:35.94

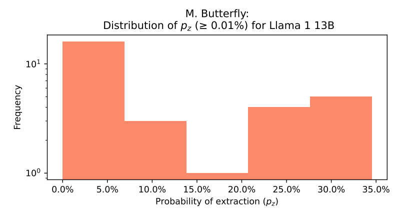
Figure 76: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



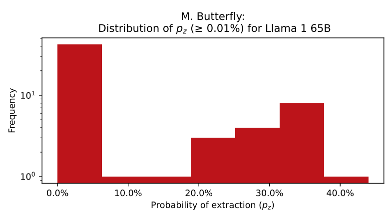
(a) PYTHIA 12B



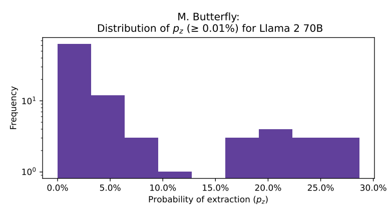
(b) PHI 4 (14B)



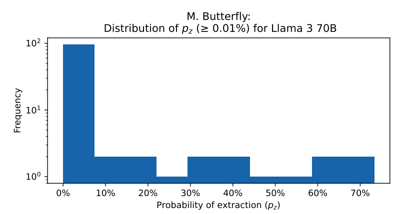
(c) LLAMA 1 13B



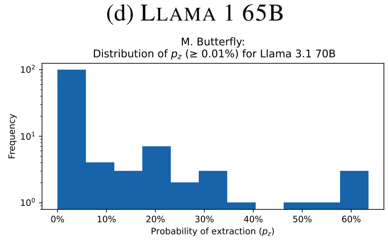
(d) LLAMA 1 65B



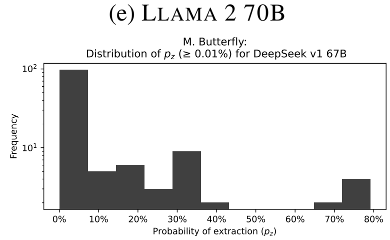
(e) LLAMA 2 70B



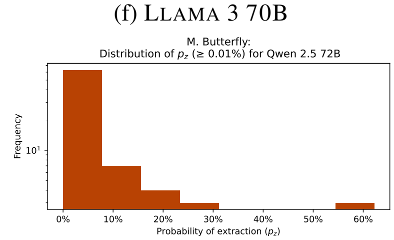
(f) LLAMA 3 70B



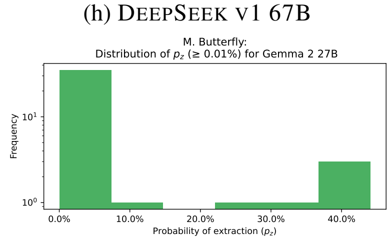
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 77: *M. Butterfly*, Hwang (1988): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

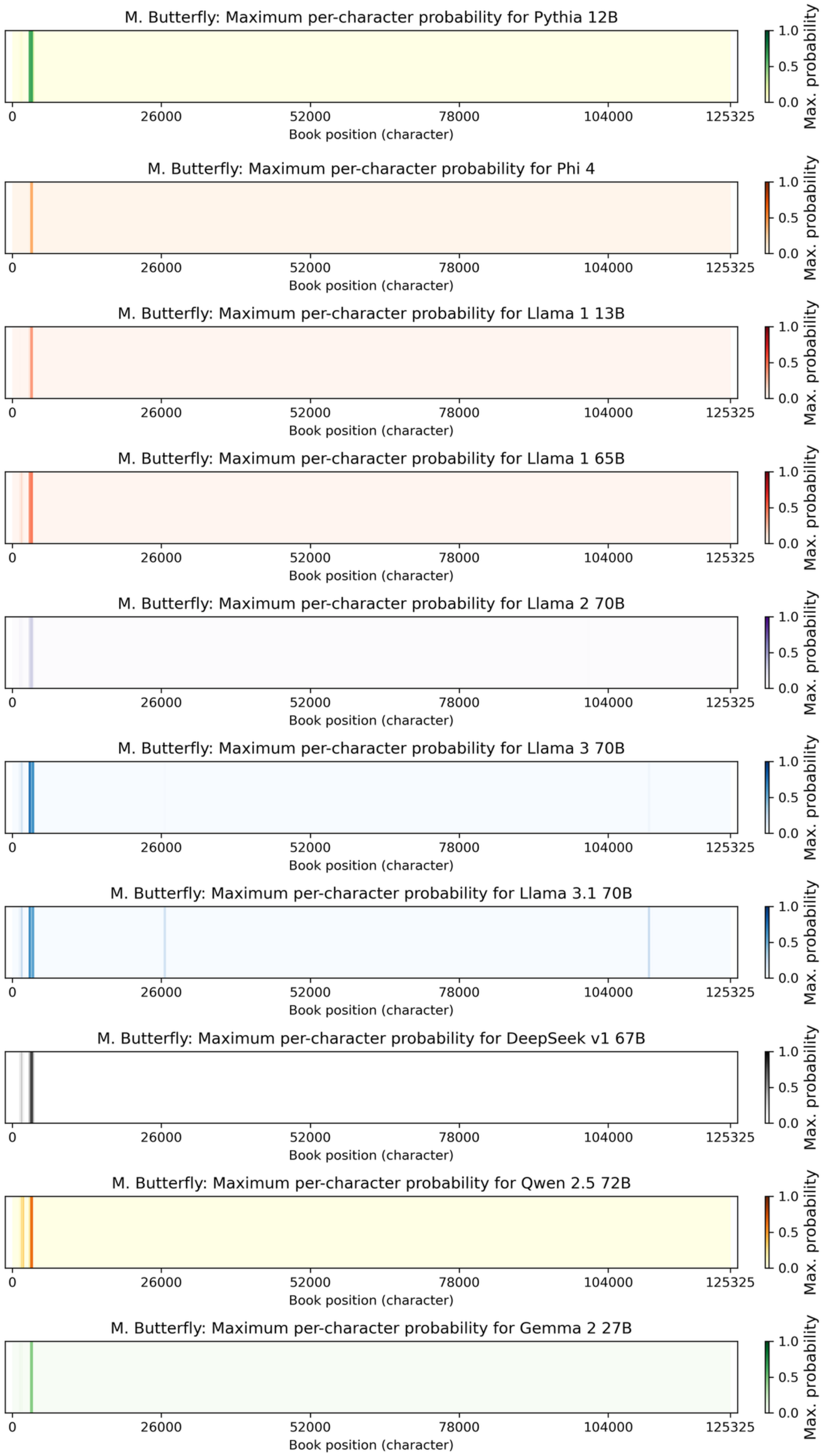
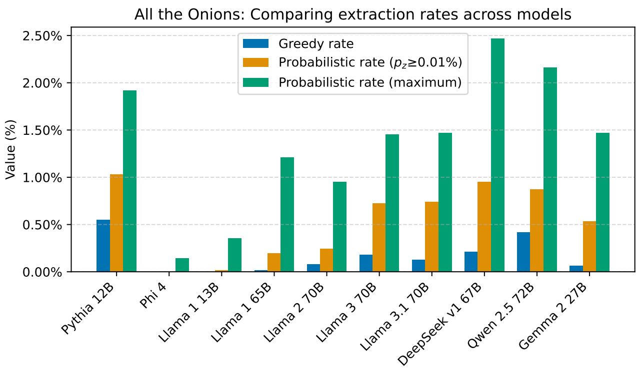


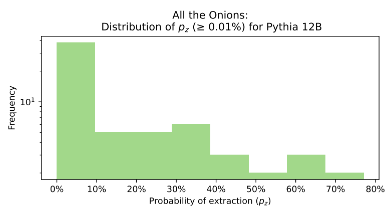
Figure 78: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.26. *All the Onions*, JACOBS

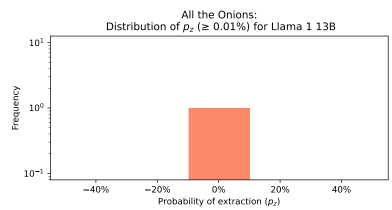


All the Onions				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.55%	1.03%	1.92%	00:02:02.96
PHI 4	0.00%	0.00%	0.15%	00:07:00.32
LLAMA 1 13B	0.00%	0.02%	0.35%	00:02:08.65
LLAMA 1 65B	0.02%	0.19%	1.21%	00:07:37.66
LLAMA 2 70B	0.08%	0.24%	0.95%	00:07:28.70
LLAMA 3 70B	0.18%	0.73%	1.45%	00:09:17.09
LLAMA 3.1 70B	0.13%	0.74%	1.47%	00:08:58.09
DEEPSEEK V1 67B	0.21%	0.95%	2.47%	00:23:04.91
QWEN 2.5 72B	0.42%	0.87%	2.16%	00:09:31.26
GEMMA 2 27B	0.06%	0.53%	1.47%	00:18:18.31

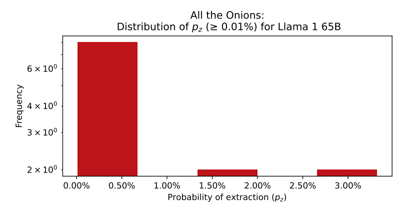
Figure 79: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



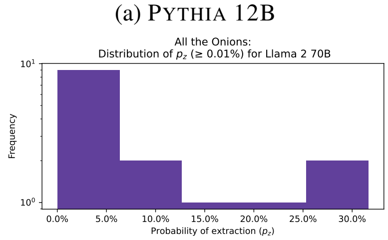
(a) PYTHIA 12B



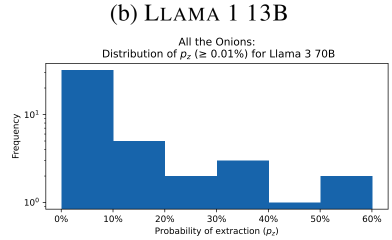
(b) LLAMA 1 13B



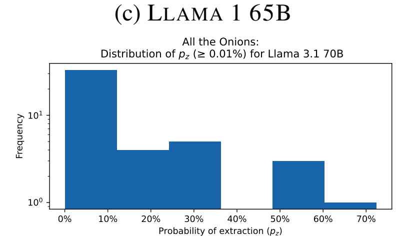
(c) LLAMA 1 65B



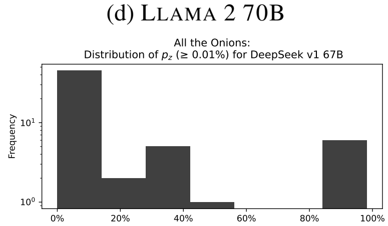
(d) LLAMA 2 70B



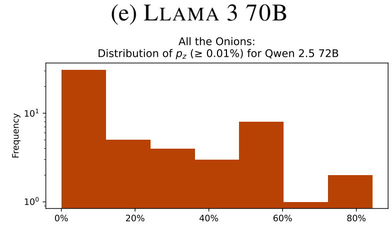
(e) LLAMA 3 70B



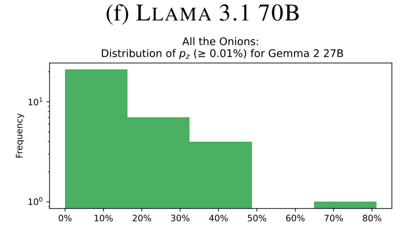
(f) LLAMA 3.1 70B



(g) DEEPSEEK V1 67B



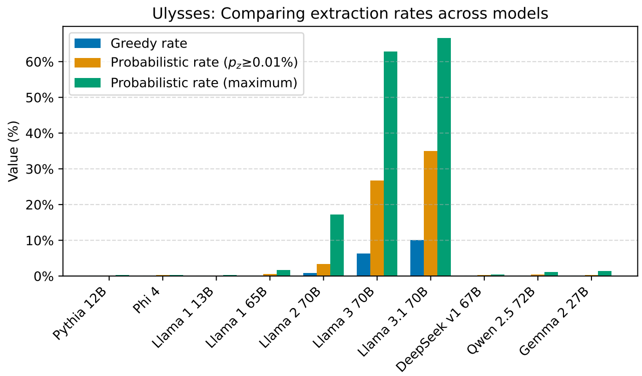
(h) QWEN 2.5 72B



(i) GEMMA 2 27B

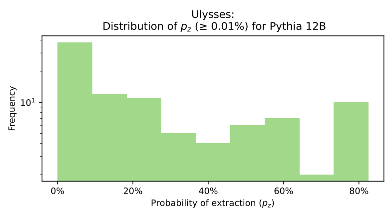
Figure 80: *All the Onions*, Jacobs (1999): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Only showing models for which there exist any $p_{\boldsymbol{z}} \geq 0.01\%$.

F.1.27. *Ulysses*, JOYCE

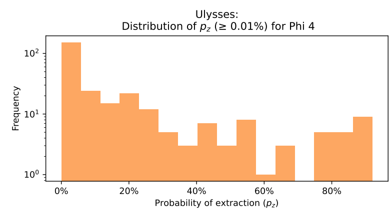


Ulysses				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.03%	0.06%	0.17%	00:52:46.60
PHI 4	0.07%	0.16%	0.29%	01:21:46.13
LLAMA 1 13B	0.03%	0.11%	0.28%	00:53:08.41
LLAMA 1 65B	0.13%	0.50%	1.59%	02:55:55.54
LLAMA 2 70B	0.75%	3.29%	17.20%	02:57:13.25
LLAMA 3 70B	6.30%	26.64%	62.76%	03:33:22.11
LLAMA 3.1 70B	10.01%	34.98%	66.53%	03:33:04.45
DEEPSEEK V1 67B	0.07%	0.22%	0.42%	03:40:03.91
QWEN 2.5 72B	0.12%	0.36%	1.05%	04:11:40.60
GEMMA 2 27B	0.06%	0.26%	1.33%	03:19:29.68

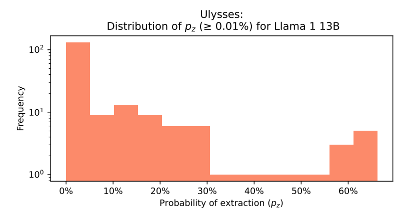
Figure 81: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



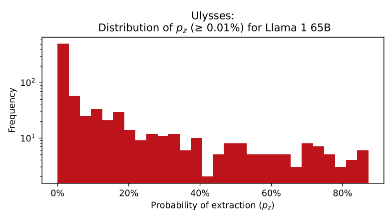
(a) PYTHIA 12B



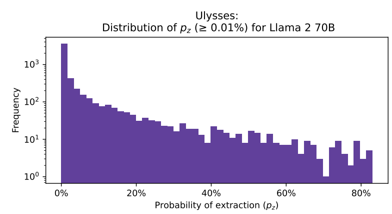
(b) PHI 4 (14B)



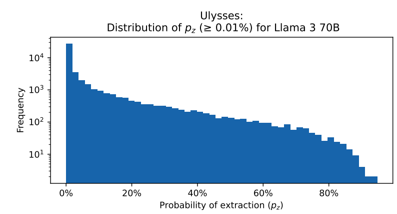
(c) LLAMA 1 13B



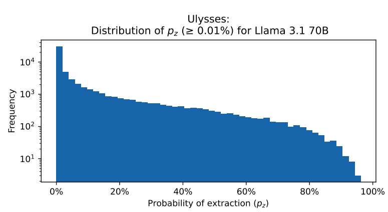
(d) LLAMA 1 65B



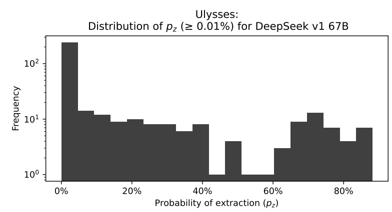
(e) LLAMA 2 70B



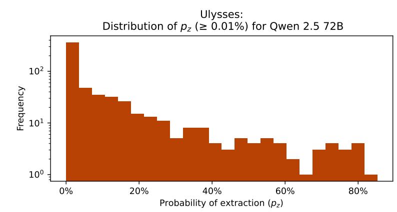
(f) LLAMA 3 70B



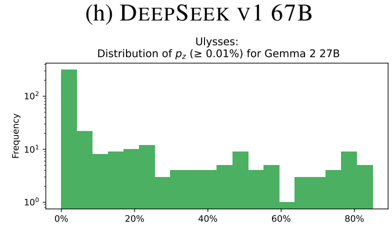
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 82: *Ulysses*, Joyce (1922): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models.

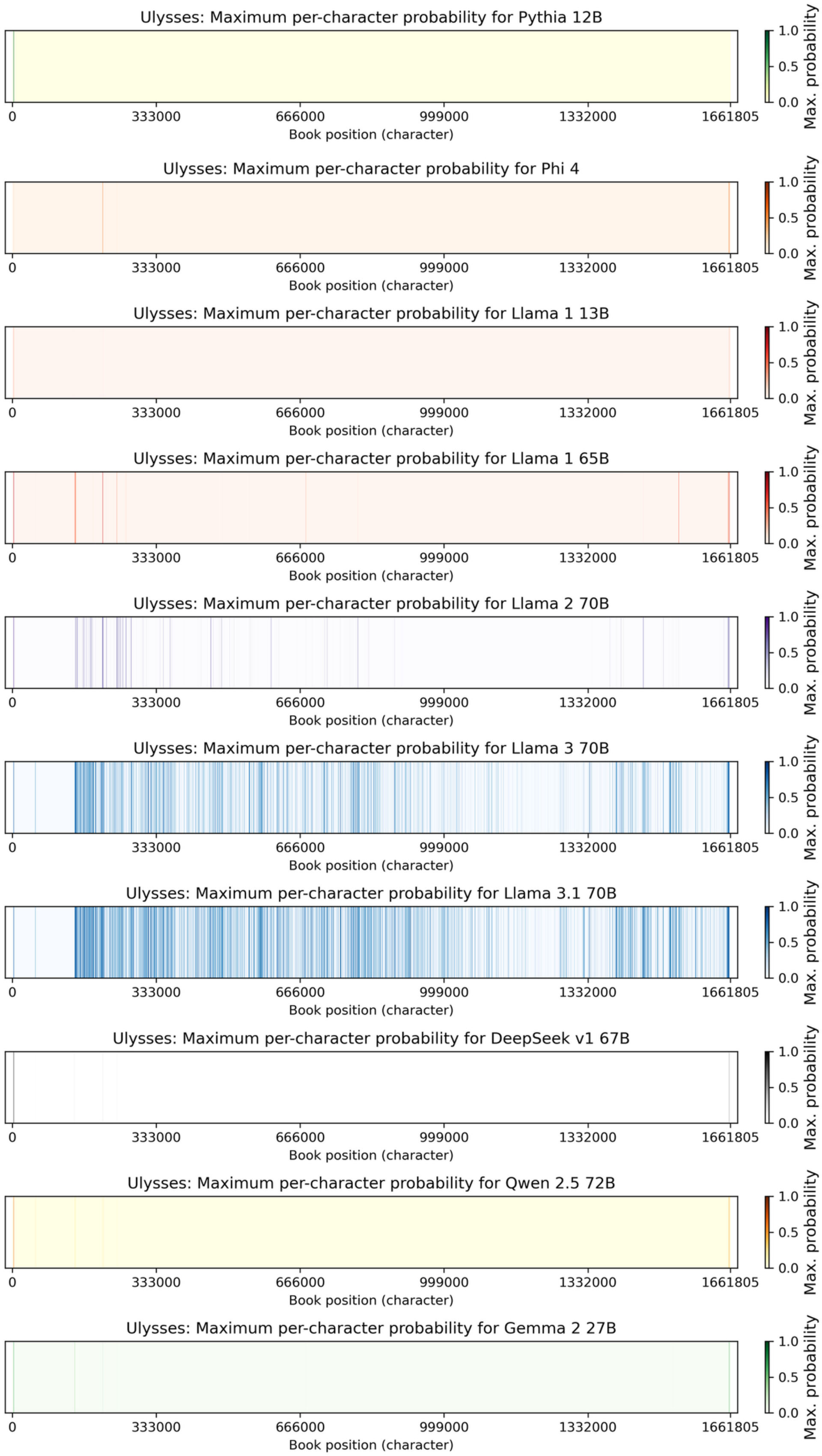
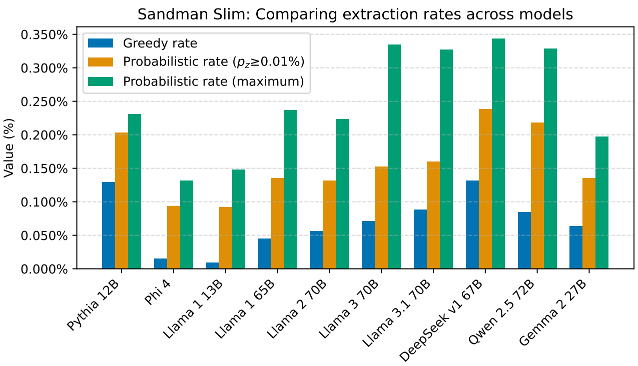


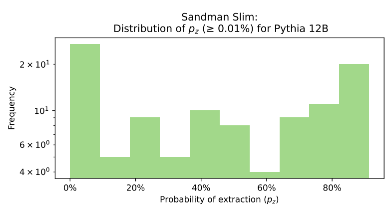
Figure 83: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.28. *Sandman Slim*, KADREY

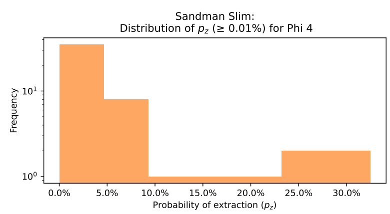


		Sandman Slim		
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.13%	0.20%	0.23%	00:35:55.95
PHI 4	0.02%	0.09%	0.13%	01:09:51.93
LLAMA 1 13B	0.01%	0.09%	0.15%	00:25:07.35
LLAMA 1 65B	0.05%	0.14%	0.24%	01:07:04.32
LLAMA 2 70B	0.06%	0.13%	0.22%	00:56:59.89
LLAMA 3 70B	0.07%	0.15%	0.33%	01:09:59.60
LLAMA 3.1 70B	0.09%	0.16%	0.33%	01:10:34.01
DEEPSEEK V1 67B	0.13%	0.24%	0.34%	01:20:56.42
QWEN 2.5 72B	0.08%	0.22%	0.33%	01:28:06.86
GEMMA 2 27B	0.06%	0.14%	0.20%	01:14:49.98

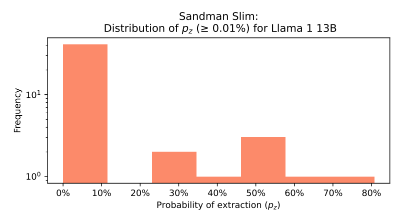
Figure 84: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



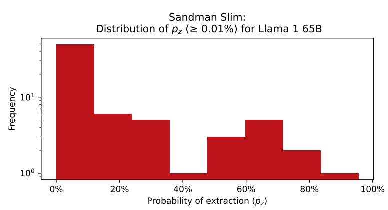
(a) PYTHIA 12B



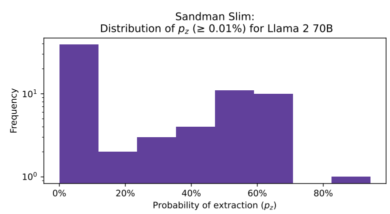
(b) PHI 4 (14B)



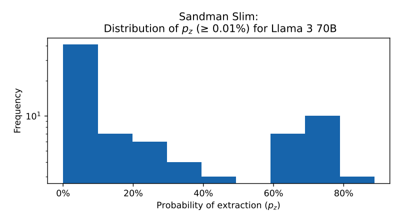
(c) LLAMA 1 13B



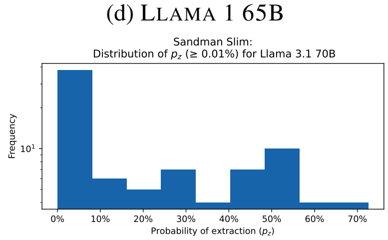
(d) LLAMA 1 65B



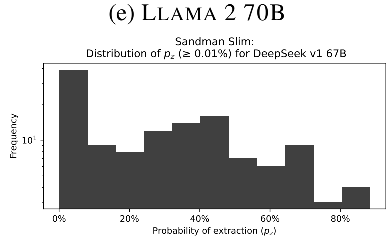
(e) LLAMA 2 70B



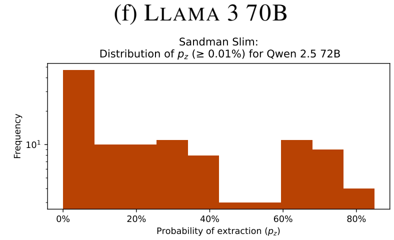
(f) LLAMA 3 70B



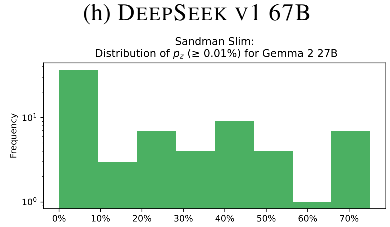
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



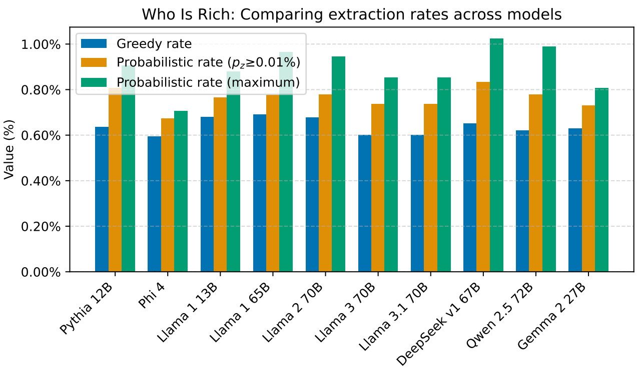
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

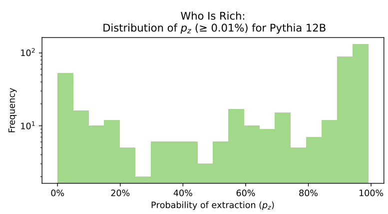
Figure 85: *Sandman Slim*, Kadrey (2009): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.29. *Who Is Rich*, KLAM

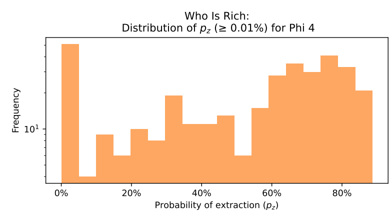


Who Is Rich				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.64%	0.81%	0.90%	00:33:56.93
PHI 4	0.59%	0.67%	0.71%	01:07:13.60
LLAMA 1 13B	0.68%	0.76%	0.88%	00:23:58.84
LLAMA 1 65B	0.69%	0.78%	0.96%	01:05:50.46
LLAMA 2 70B	0.68%	0.78%	0.94%	00:56:11.63
LLAMA 3 70B	0.60%	0.74%	0.85%	01:07:24.48
LLAMA 3.1 70B	0.60%	0.74%	0.85%	01:06:56.61
DEEPSEEK V1 67B	0.65%	0.83%	1.02%	01:17:45.64
QWEN 2.5 72B	0.62%	0.78%	0.99%	01:30:40.41
GEMMA 2 27B	0.63%	0.73%	0.81%	01:16:21.60

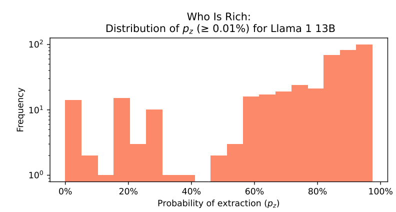
Figure 86: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



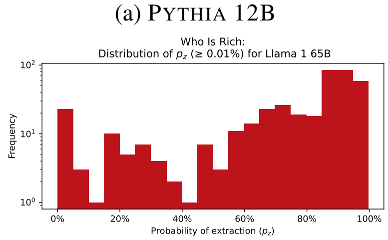
(a) PYTHIA 12B



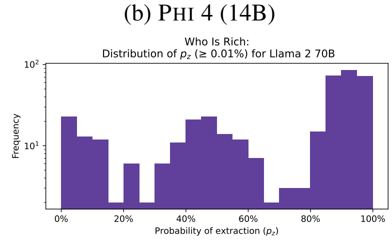
(b) PHI 4 (14B)



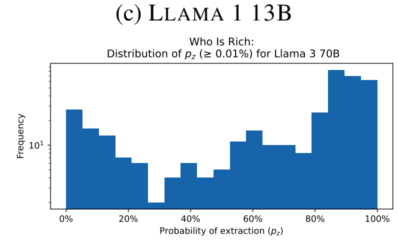
(c) LLAMA 1 13B



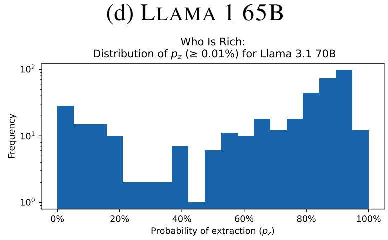
(d) LLAMA 1 65B



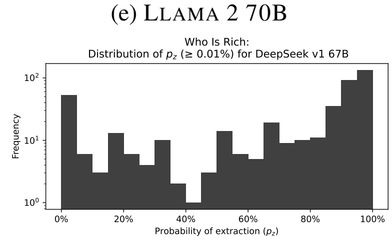
(e) LLAMA 2 70B



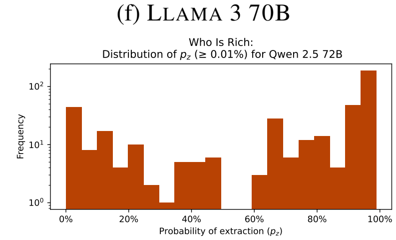
(f) LLAMA 3 70B



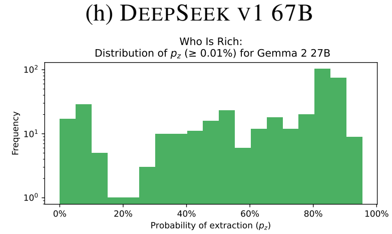
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



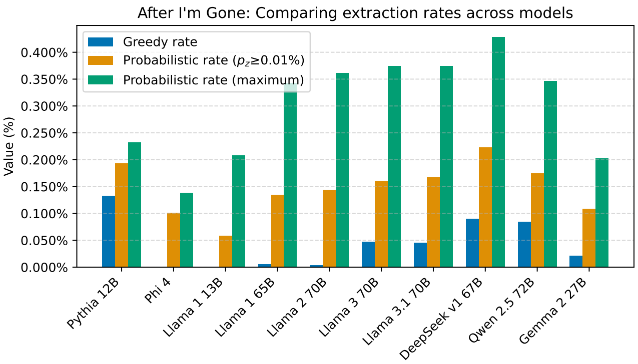
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

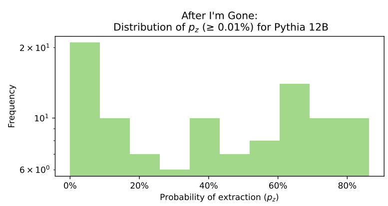
Figure 87: *Who Is Rich*, Klam (2017): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.30. *After I'm Gone*, LIPPMAN

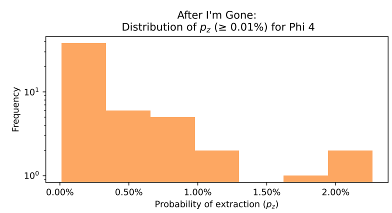


After I'm Gone				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.13%	0.19%	0.23%	00:33:12.10
PHI 4	0.00%	0.10%	0.14%	01:06:48.44
LLAMA 1 13B	0.00%	0.06%	0.21%	00:24:18.57
LLAMA 1 65B	0.01%	0.13%	0.34%	01:05:17.58
LLAMA 2 70B	0.00%	0.14%	0.36%	00:57:47.55
LLAMA 3 70B	0.05%	0.16%	0.37%	01:10:27.58
LLAMA 3.1 70B	0.04%	0.17%	0.37%	01:09:50.46
DEEPSEEK V1 67B	0.09%	0.22%	0.43%	01:18:42.58
QWEN 2.5 72B	0.08%	0.17%	0.35%	01:28:59.07
GEMMA 2 27B	0.02%	0.11%	0.20%	01:07:40.31

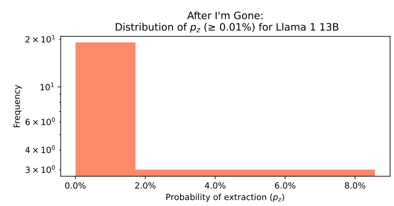
Figure 88: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



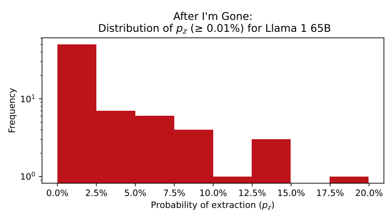
(a) PYTHIA 12B



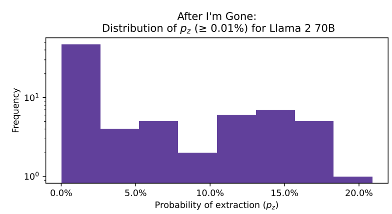
(b) PHI 4 (14B)



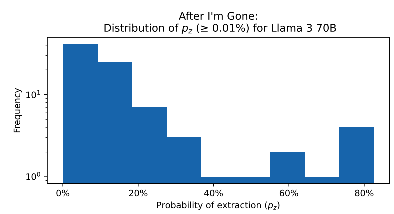
(c) LLAMA 1 13B



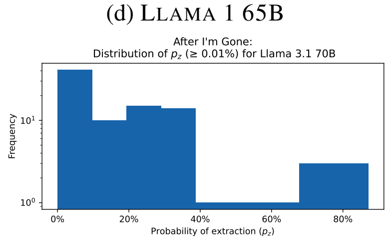
(d) LLAMA 1 65B



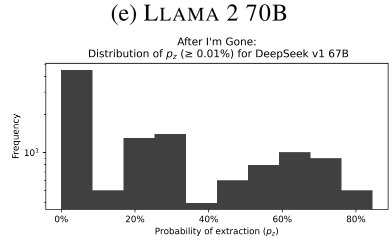
(e) LLAMA 2 70B



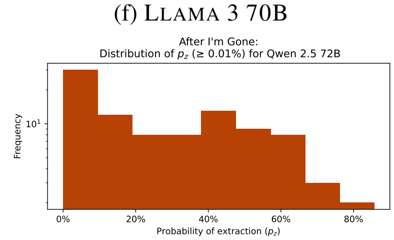
(f) LLAMA 3 70B



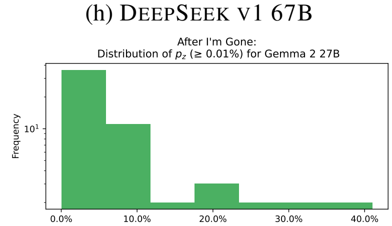
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



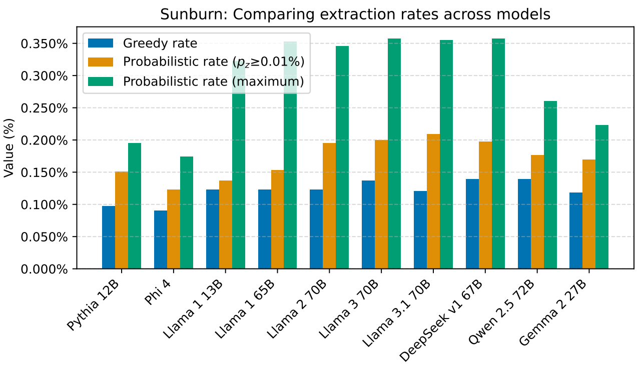
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

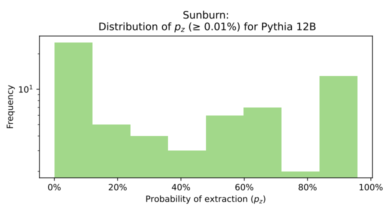
Figure 89: *After I'm Gone*, Lippman (2014): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.31. *Sunburn*, LIPPMAN

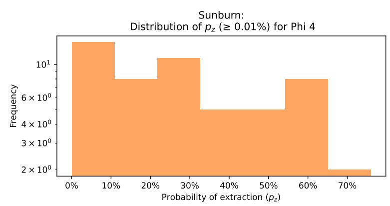


Sunburn				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.10%	0.15%	0.20%	00:14:07.97
PHI 4	0.09%	0.12%	0.17%	00:24:21.27
LLAMA 1 13B	0.12%	0.14%	0.32%	00:13:09.52
LLAMA 1 65B	0.12%	0.15%	0.35%	00:47:04.89
LLAMA 2 70B	0.12%	0.20%	0.35%	00:46:16.38
LLAMA 3 70B	0.14%	0.20%	0.36%	00:56:19.30
LLAMA 3.1 70B	0.12%	0.21%	0.36%	00:57:14.94
DEEPSEEK V1 67B	0.14%	0.20%	0.36%	01:08:03.62
QWEN 2.5 72B	0.14%	0.18%	0.26%	01:14:02.46
GEMMA 2 27B	0.12%	0.17%	0.22%	01:01:05.27

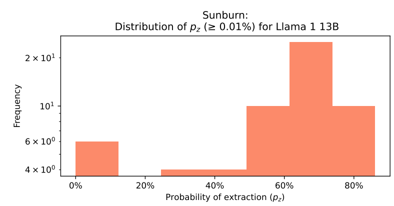
Figure 90: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



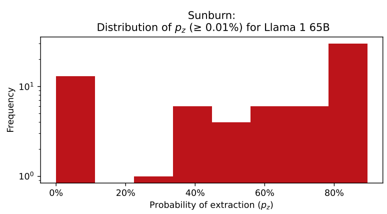
(a) PYTHIA 12B



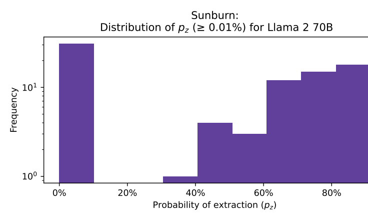
(b) PHI 4 (14B)



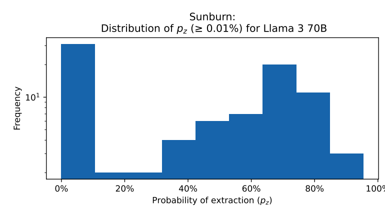
(c) LLAMA 1 13B



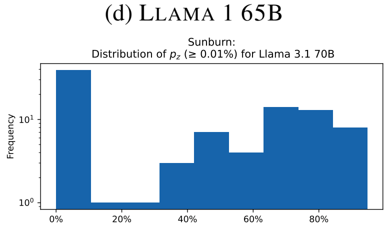
(d) LLAMA 1 65B



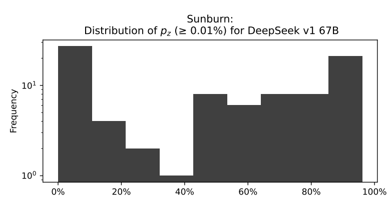
(e) LLAMA 2 70B



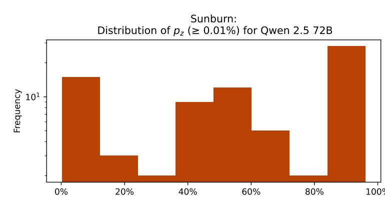
(f) LLAMA 3 70B



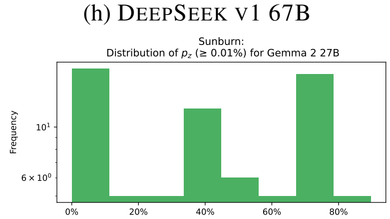
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



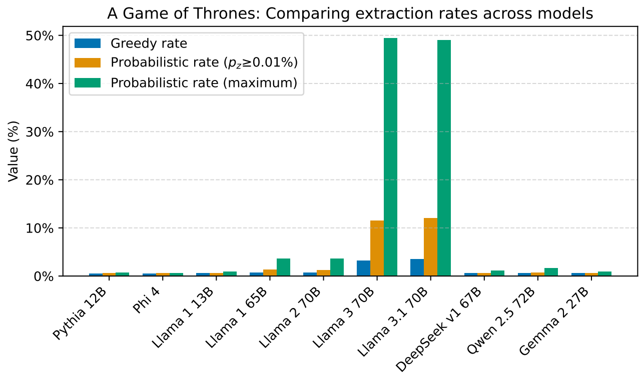
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

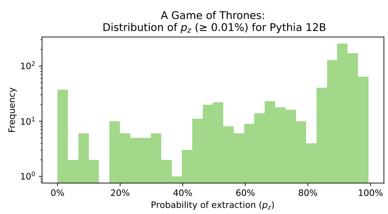
Figure 91: *Sunburn*, Lippman (2018): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models.

F.1.32. *A Game of Thrones*, MARTIN

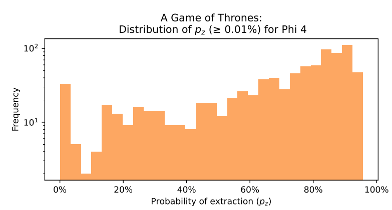


A Game of Thrones				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.52%	0.55%	0.70%	00:50:39.51
PHI 4	0.51%	0.54%	0.63%	01:25:11.53
LLAMA 1 13B	0.54%	0.58%	0.85%	00:50:41.26
LLAMA 1 65B	0.73%	1.29%	3.64%	02:52:01.23
LLAMA 2 70B	0.70%	1.19%	3.65%	02:58:28.25
LLAMA 3 70B	3.21%	11.54%	49.39%	03:35:35.17
LLAMA 3.1 70B	3.53%	11.97%	49.00%	03:35:27.98
DEEPSEEK V1 67B	0.55%	0.62%	1.14%	03:33:39.71
QWEN 2.5 72B	0.55%	0.71%	1.65%	03:57:53.28
GEMMA 2 27B	0.54%	0.60%	0.88%	03:15:31.02

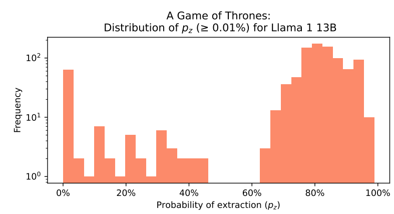
Figure 92: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



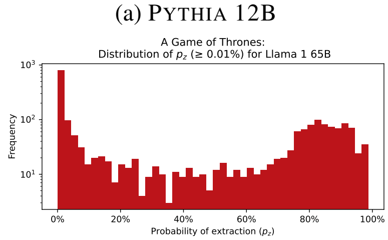
(a) PYTHIA 12B



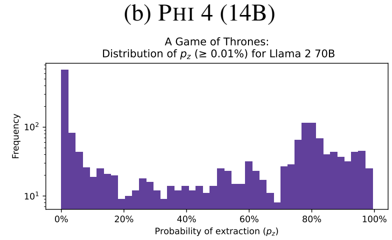
(b) PHI 4 (14B)



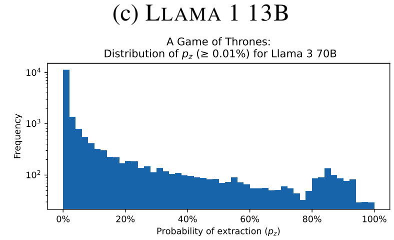
(c) LLAMA 1 13B



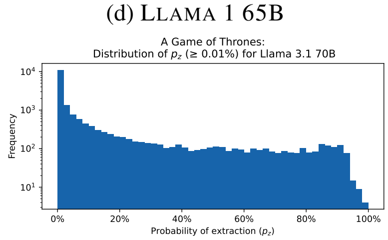
(d) LLAMA 1 65B



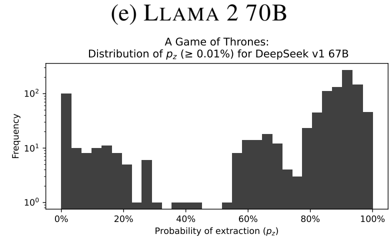
(e) LLAMA 2 70B



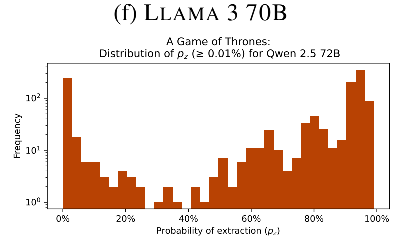
(f) LLAMA 3 70B



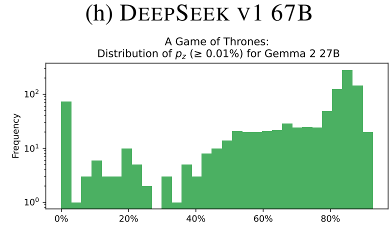
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 93: *A Game of Thrones*, Martin (1996): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

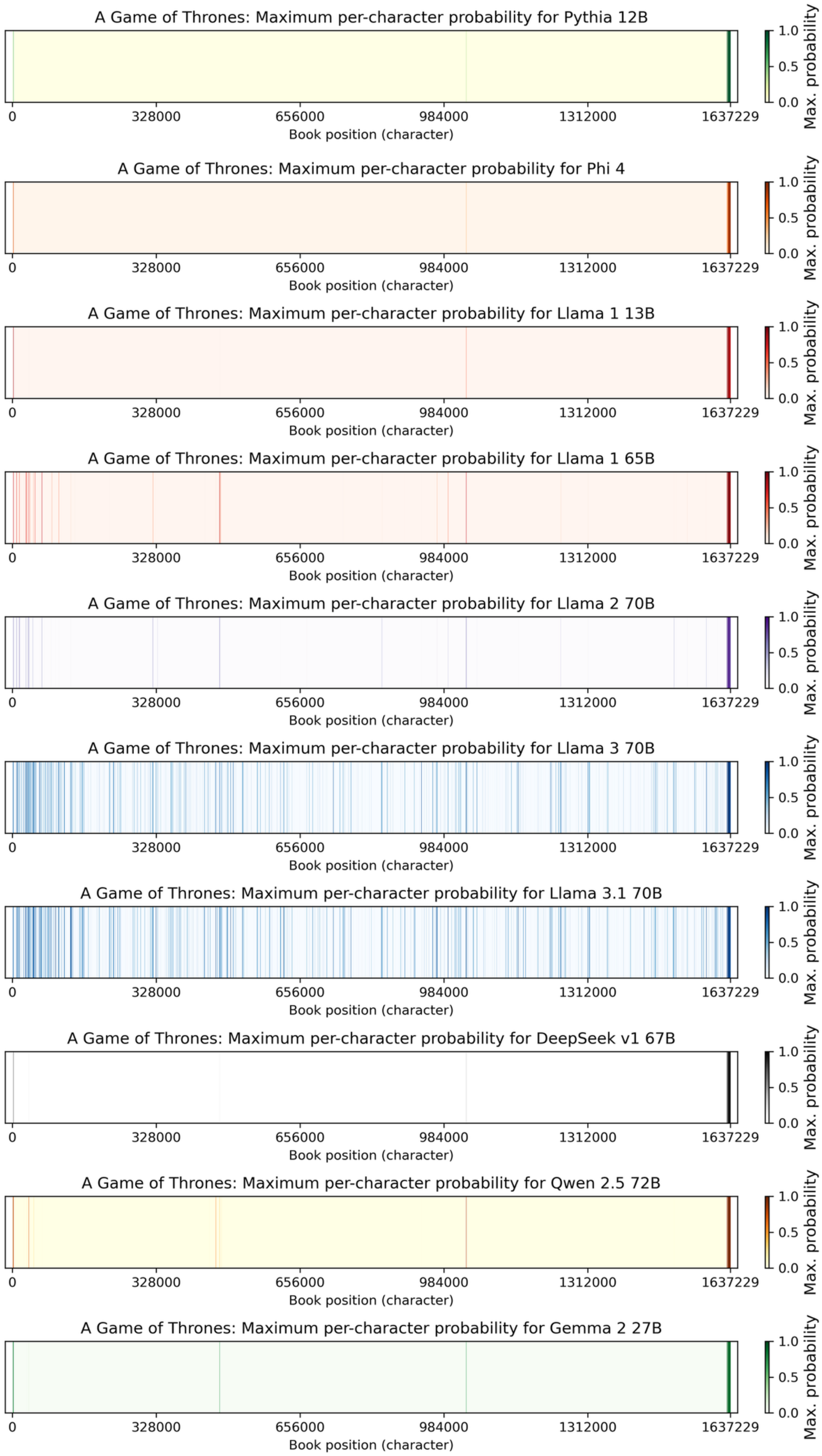
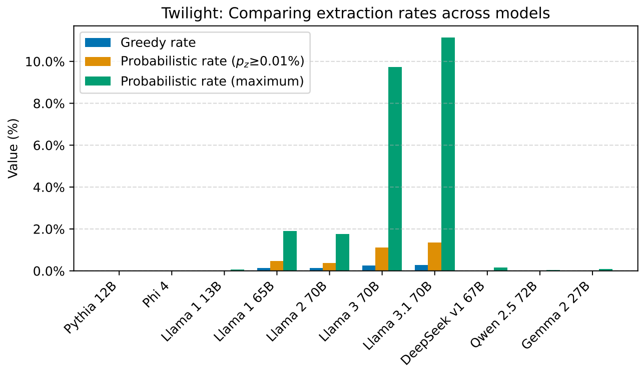


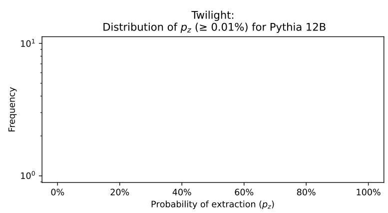
Figure 94: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.33. *Twilight*, MEYER

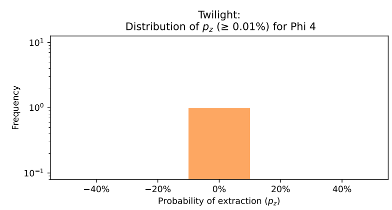


		Twilight		
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.00%	0.00%	00:21:00.06
PHI 4	0.00%	0.00%	0.01%	00:34:48.12
LLAMA 1 13B	0.00%	0.00%	0.07%	00:23:01.77
LLAMA 1 65B	0.13%	0.46%	1.90%	01:13:54.80
LLAMA 2 70B	0.12%	0.36%	1.76%	01:15:17.73
LLAMA 3 70B	0.24%	1.11%	9.73%	01:30:38.57
LLAMA 3.1 70B	0.27%	1.35%	11.14%	01:31:17.49
DEEPSEEK V1 67B	0.00%	0.00%	0.14%	01:39:12.76
QWEN 2.5 72B	0.00%	0.00%	0.03%	01:54:47.95
GEMMA 2 27B	0.00%	0.02%	0.08%	01:34:18.93

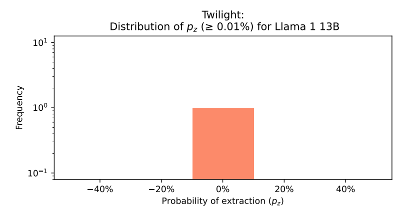
Figure 95: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



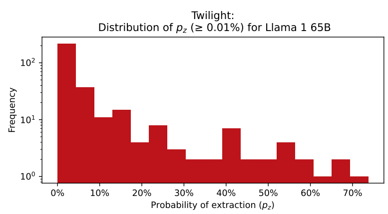
(a) PYTHIA 12B



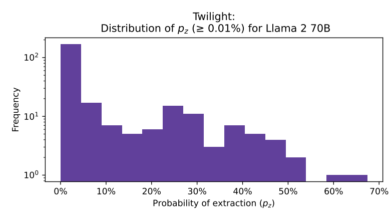
(b) PHI 4 (14B)



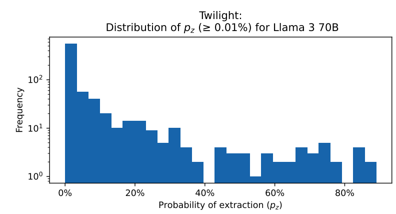
(c) LLAMA 1 13B



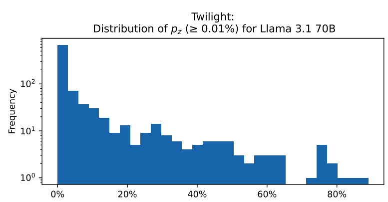
(d) LLAMA 1 65B



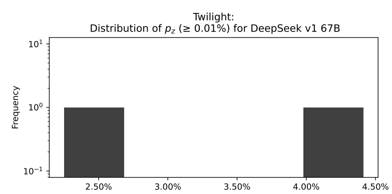
(e) LLAMA 2 70B



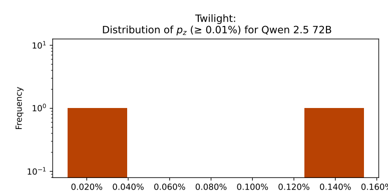
(f) LLAMA 3 70B



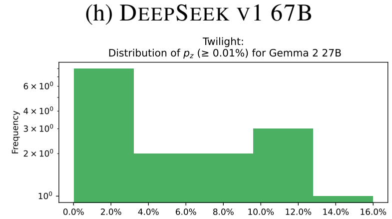
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 96: *Twilight*, Meyer (2005): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

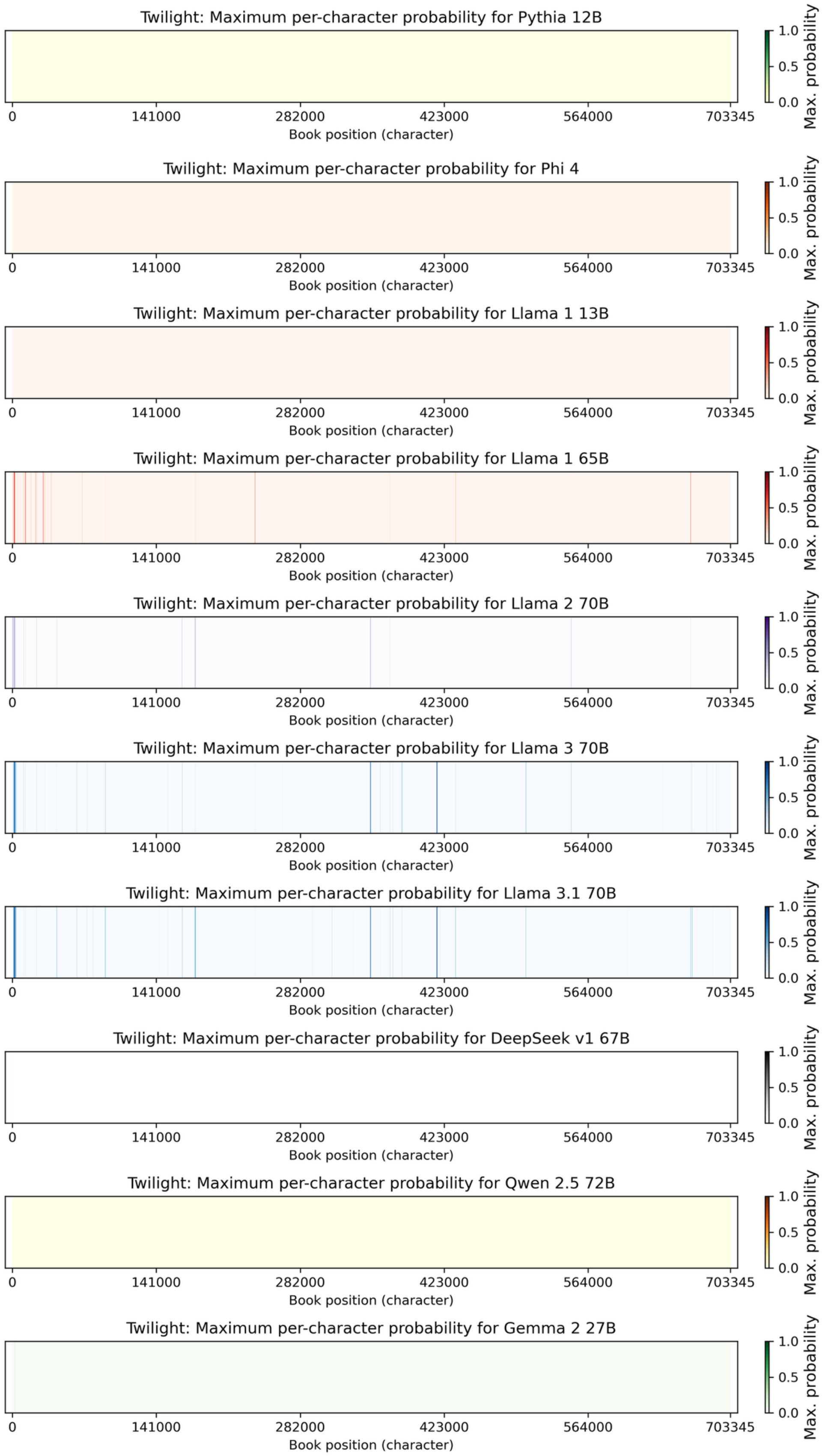
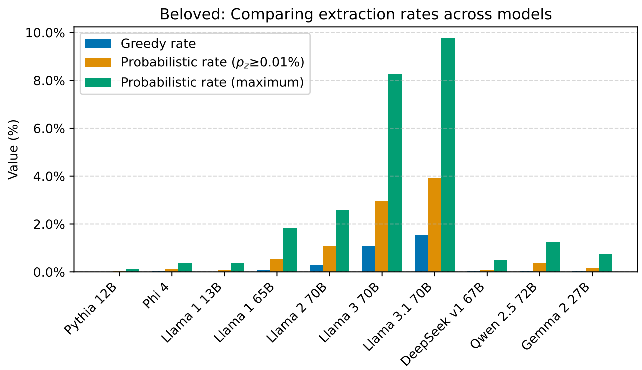


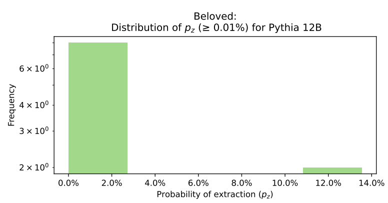
Figure 97: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.34. *Beloved*, MORRISON

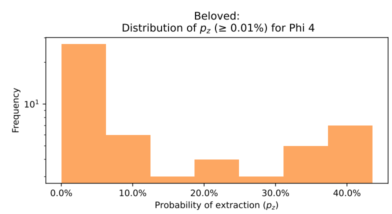


		Beloved		
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.02%	0.10%	00:15:06.06
PHI 4	0.05%	0.10%	0.35%	00:28:41.59
LLAMA 1 13B	0.00%	0.06%	0.35%	00:16:46.39
LLAMA 1 65B	0.09%	0.55%	1.84%	00:58:18.24
LLAMA 2 70B	0.27%	1.06%	2.60%	00:57:14.62
LLAMA 3 70B	1.06%	2.95%	8.26%	01:09:13.78
LLAMA 3.1 70B	1.52%	3.93%	9.75%	01:10:59.00
DEEPSEEK V1 67B	0.02%	0.08%	0.49%	01:16:38.75
QWEN 2.5 72B	0.05%	0.36%	1.23%	01:18:30.16
GEMMA 2 27B	0.01%	0.16%	0.73%	01:07:15.42

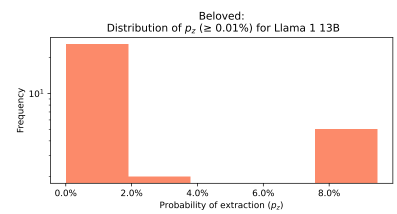
Figure 98: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



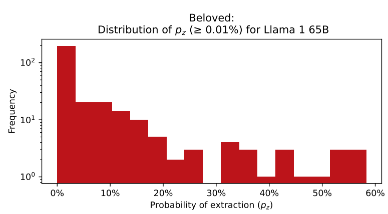
(a) PYTHIA 12B



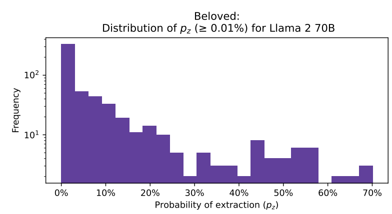
(b) PHI 4 (14B)



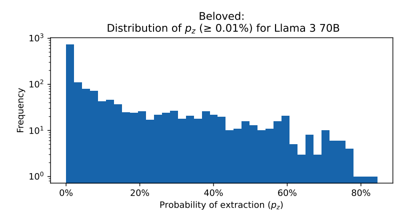
(c) LLAMA 1 13B



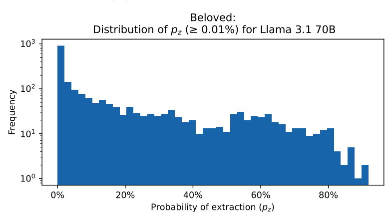
(d) LLAMA 1 65B



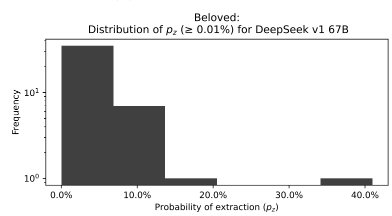
(e) LLAMA 2 70B



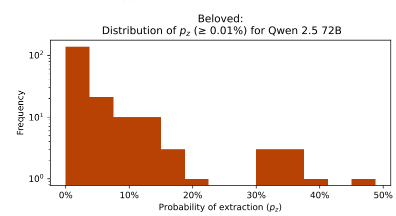
(f) LLAMA 3 70B



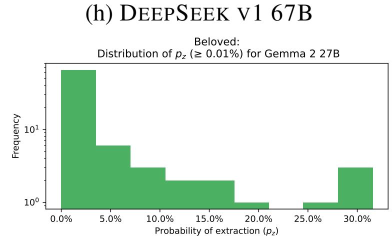
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 99: *Beloved*, Morrison (1987): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

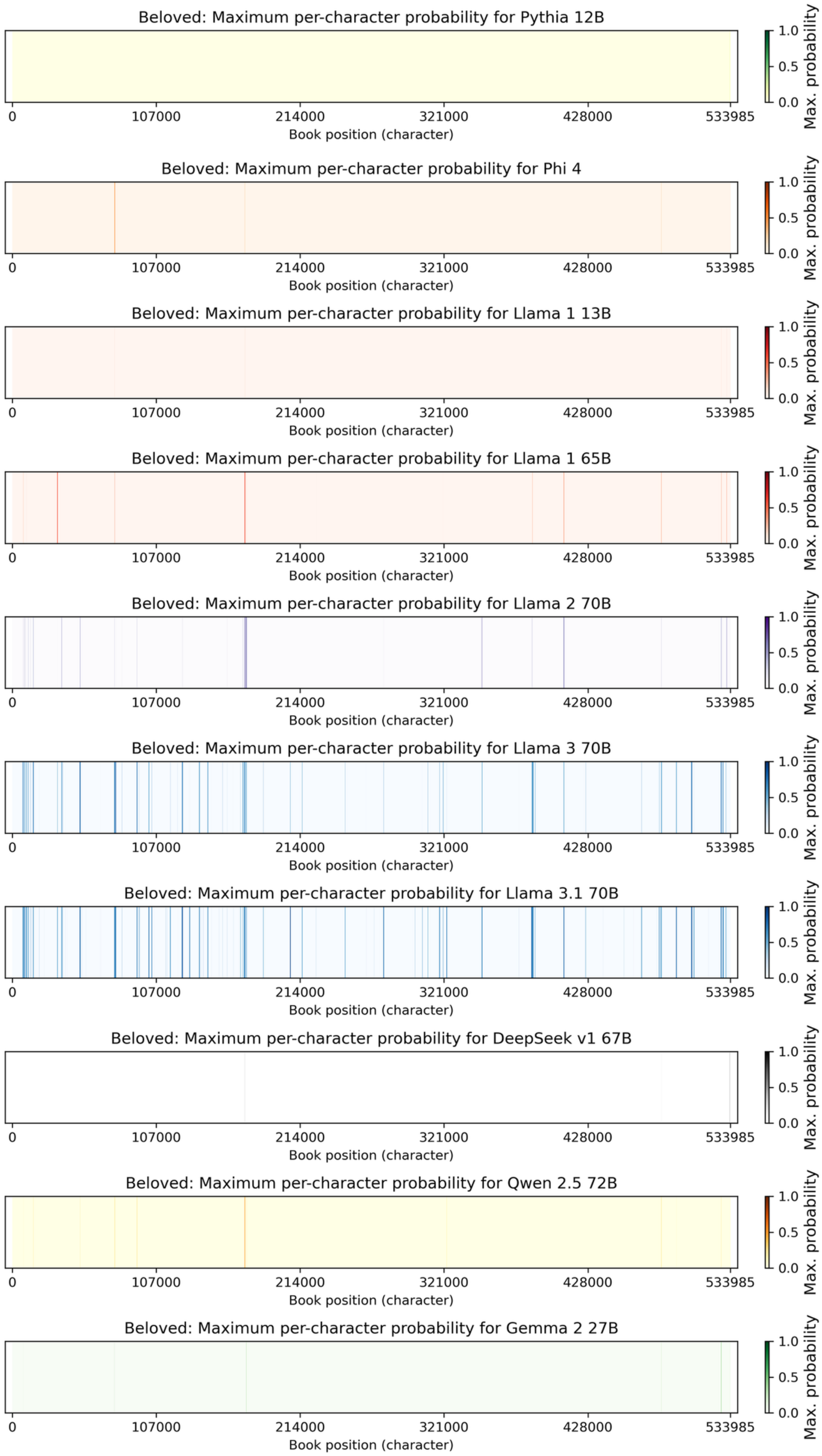
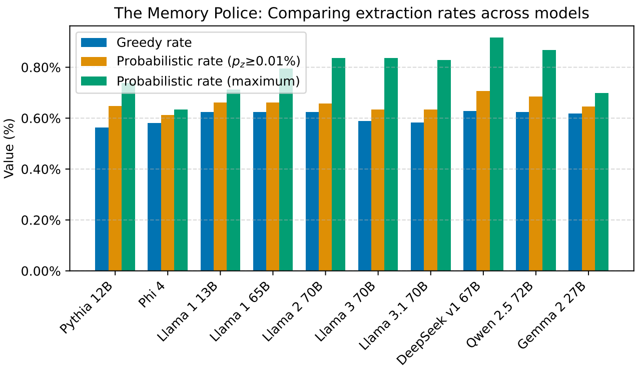


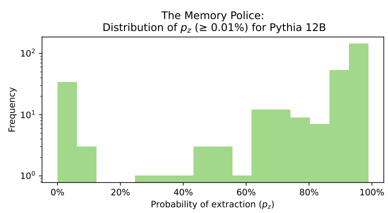
Figure 100: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.35. *The Memory Police*, OGAWA

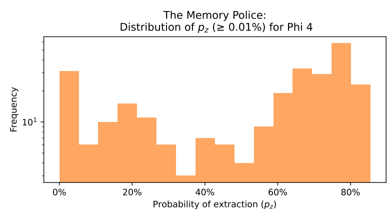


The Memory Police				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.56%	0.65%	0.74%	00:13:17.65
PHI 4	0.58%	0.61%	0.63%	00:25:00.18
LLAMA 1 13B	0.62%	0.66%	0.71%	00:13:25.28
LLAMA 1 65B	0.62%	0.66%	0.79%	00:49:08.03
LLAMA 2 70B	0.62%	0.66%	0.84%	00:47:45.59
LLAMA 3 70B	0.59%	0.63%	0.84%	00:59:09.73
LLAMA 3.1 70B	0.58%	0.63%	0.83%	00:57:40.92
DEEPSEEK V1 67B	0.63%	0.71%	0.92%	01:06:15.23
QWEN 2.5 72B	0.62%	0.69%	0.87%	01:01:38.03
GEMMA 2 27B	0.62%	0.64%	0.70%	01:01:52.56

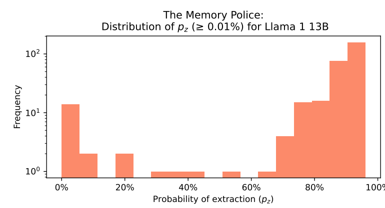
Figure 101: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



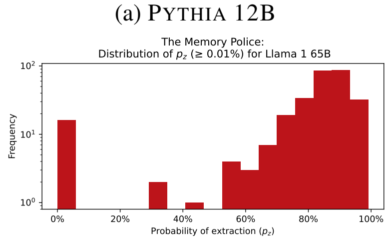
(a) PYTHIA 12B



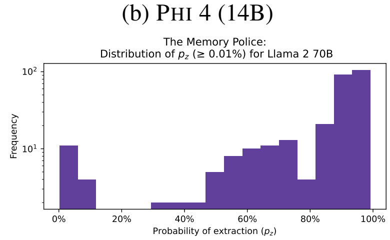
(b) PHI 4 (14B)



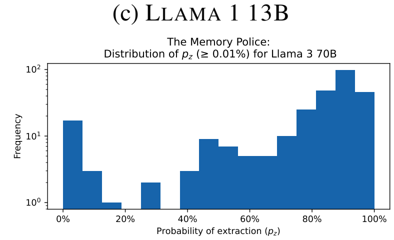
(c) LLAMA 1 13B



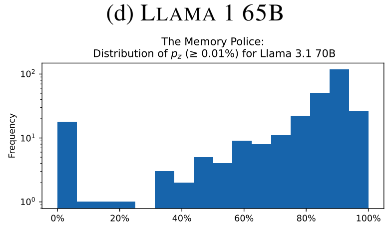
(d) LLAMA 1 65B



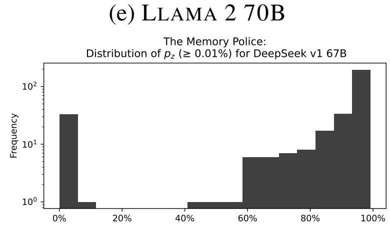
(e) LLAMA 2 70B



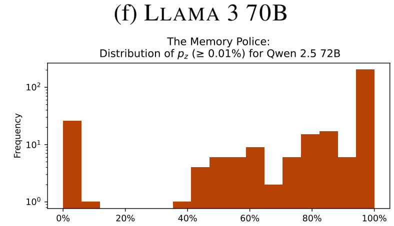
(f) LLAMA 3 70B



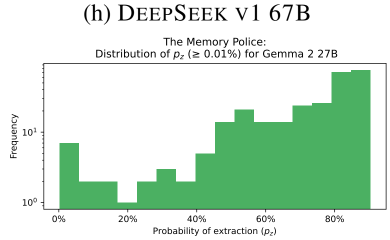
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



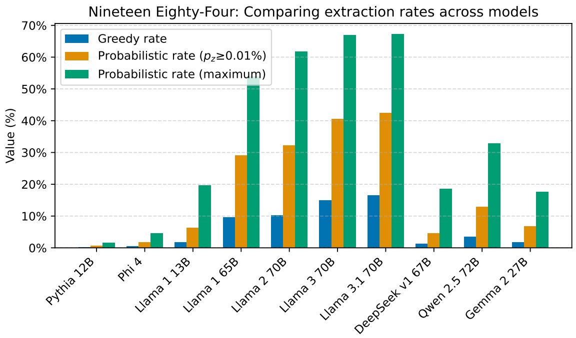
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

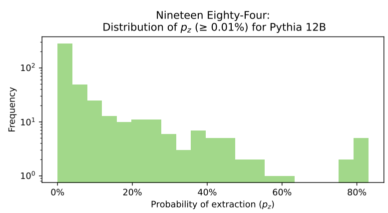
Figure 102: *The Memory Police*, Ogawa (2019): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.36. *Nineteen Eighty-Four*, ORWELL

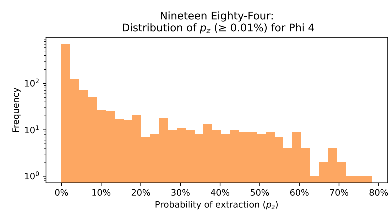


Nineteen Eighty-Four				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.22%	0.65%	1.61%	00:21:21.22
PHI 4	0.57%	1.82%	4.63%	00:35:20.85
LLAMA 1 13B	1.85%	6.39%	19.66%	00:21:23.13
LLAMA 1 65B	9.63%	29.11%	53.90%	01:11:45.83
LLAMA 2 70B	10.24%	32.26%	61.73%	01:12:58.15
LLAMA 3 70B	14.95%	40.51%	67.01%	01:29:40.53
LLAMA 3.1 70B	16.53%	42.42%	67.23%	01:31:41.77
DEEPSEEK V1 67B	1.23%	4.65%	18.57%	01:32:17.67
QWEN 2.5 72B	3.52%	12.98%	32.83%	01:41:22.32
GEMMA 2 27B	1.78%	6.80%	17.62%	01:28:44.98

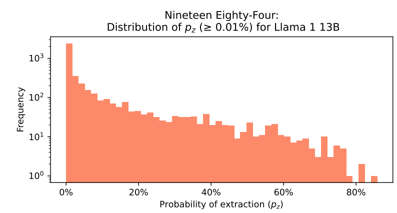
Figure 103: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



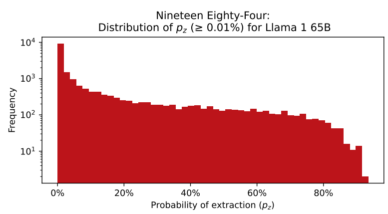
(a) PYTHIA 12B



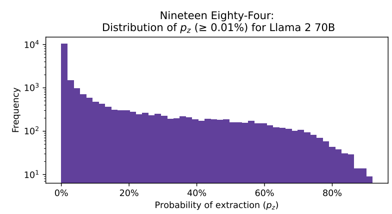
(b) PHI 4 (14B)



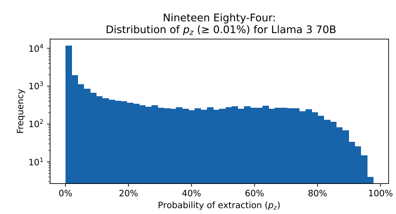
(c) LLAMA 1 13B



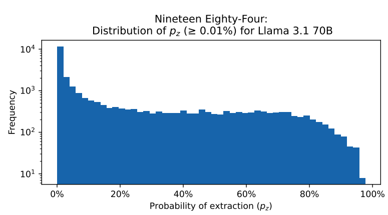
(d) LLAMA 1 65B



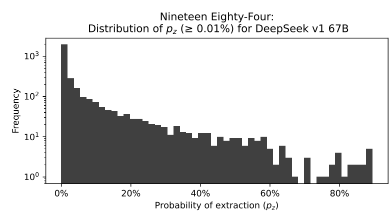
(e) LLAMA 2 70B



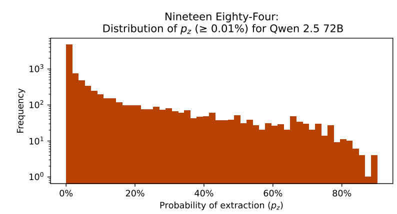
(f) LLAMA 3 70B



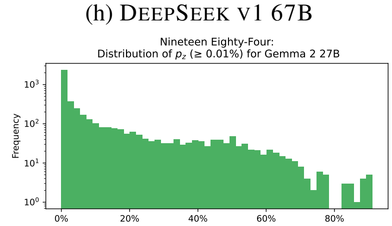
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 104: *Nineteen Eighty-Four*, Orwell (1949): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

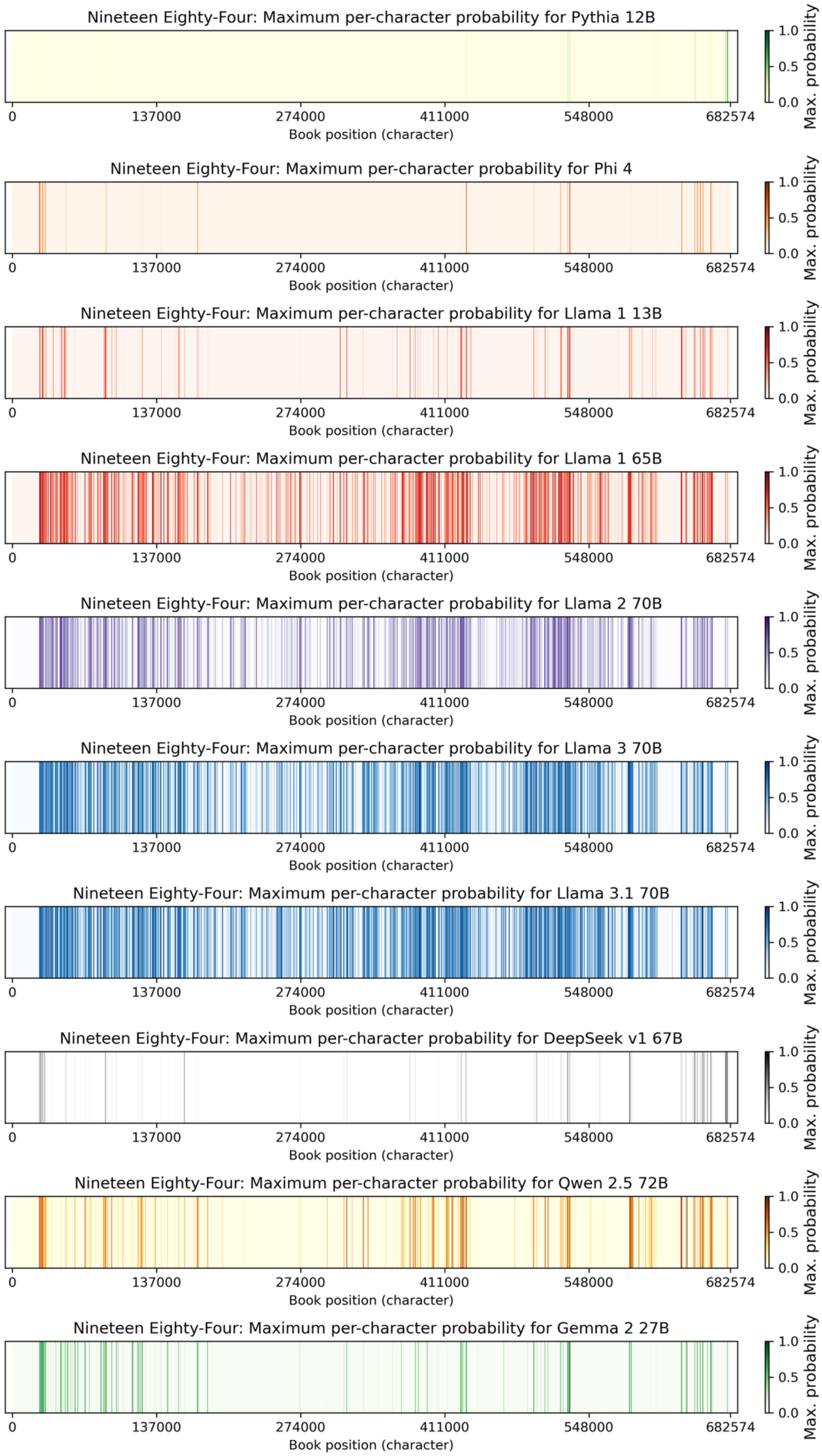
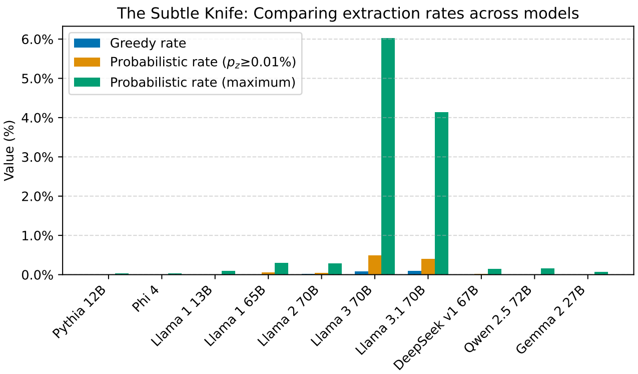


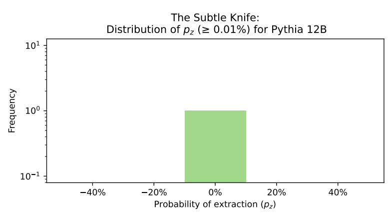
Figure 105: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.37. *The Subtle Knife*, PULLMAN

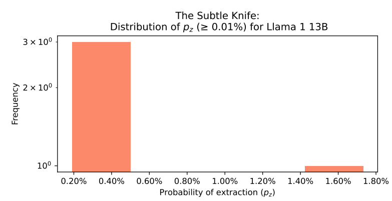


The Subtle Knife				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.00%	0.00%	0.03%	00:17:16.29
PHI 4	0.00%	0.00%	0.03%	00:28:38.11
LLAMA 1 13B	0.00%	0.01%	0.10%	00:17:24.09
LLAMA 1 65B	0.00%	0.06%	0.30%	00:59:12.84
LLAMA 2 70B	0.01%	0.05%	0.29%	02:29:51.04
LLAMA 3 70B	0.08%	0.50%	6.03%	01:11:01.69
LLAMA 3.1 70B	0.09%	0.40%	4.14%	01:11:12.10
DEEPSEEK V1 67B	0.00%	0.02%	0.14%	01:21:23.03
QWEN 2.5 72B	0.00%	0.00%	0.16%	01:15:51.04
GEMMA 2 27B	0.00%	0.00%	0.07%	01:13:01.78

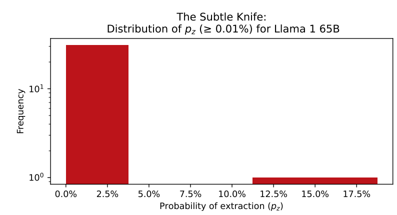
Figure 106: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



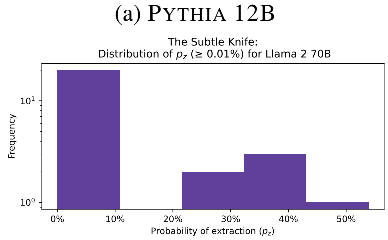
(a) PYTHIA 12B



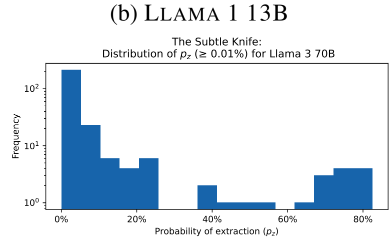
(b) LLAMA 1 13B



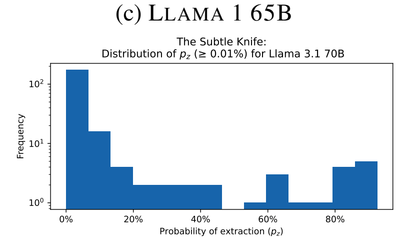
(c) LLAMA 1 65B



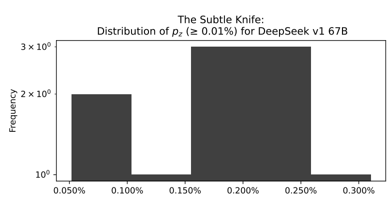
(d) LLAMA 2 70B



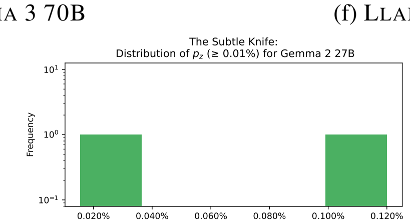
(e) LLAMA 3 70B



(f) LLAMA 3.1 70B



(g) DEEPSEEK V1 67B



(h) GEMMA 2 27B

Figure 107: *The Subtle Knife*, Pullman (1997): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Only showing models for which there exist any $p_{\boldsymbol{z}} \geq 0.01\%$.

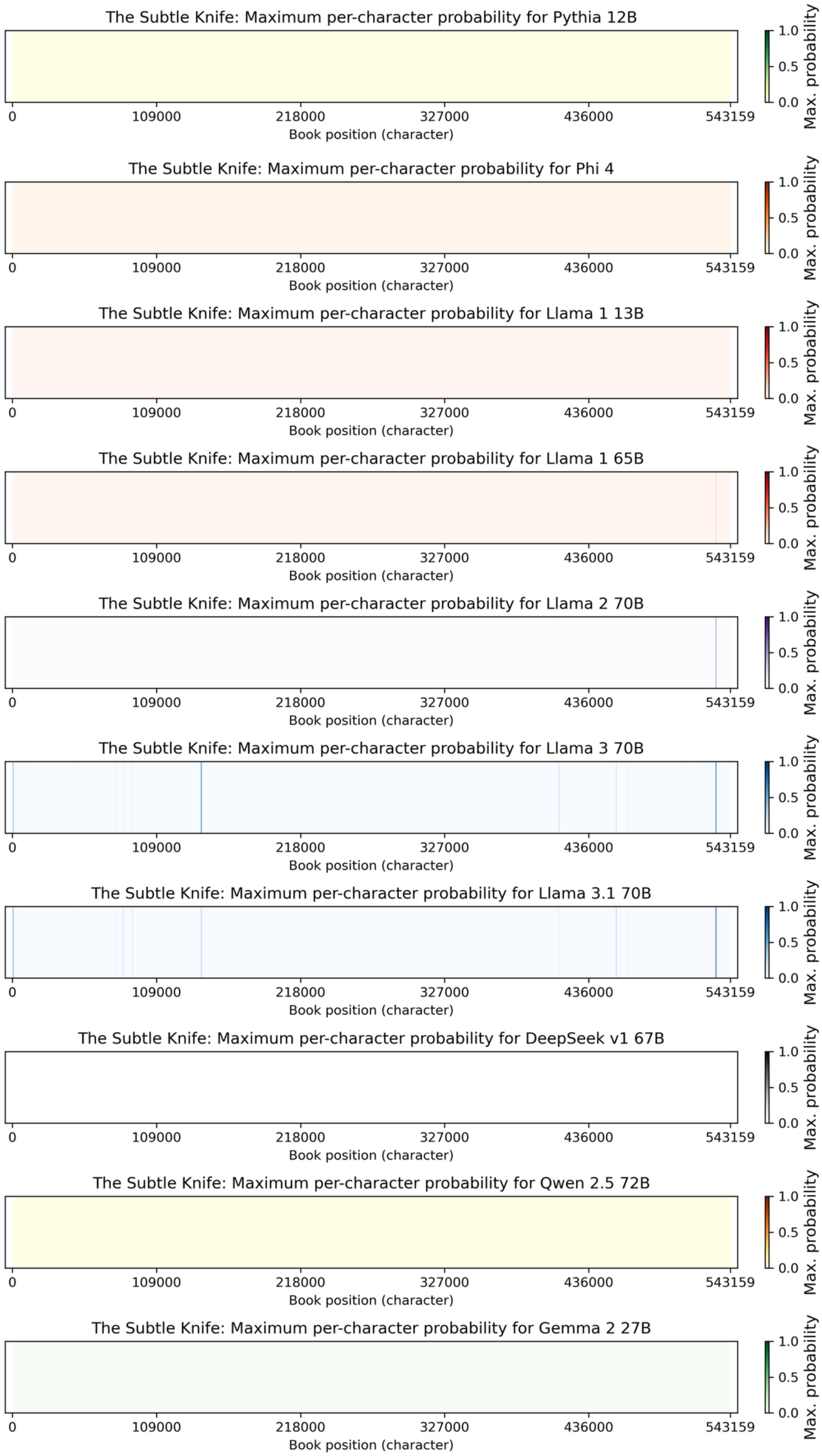
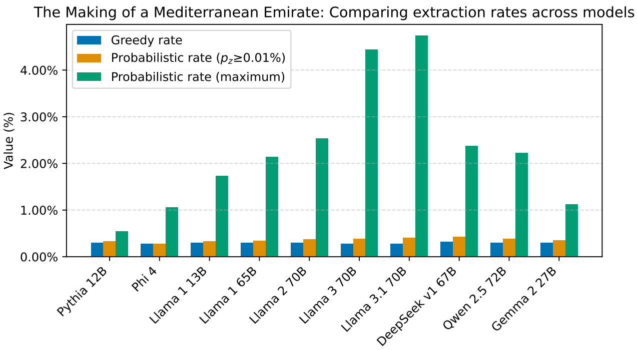


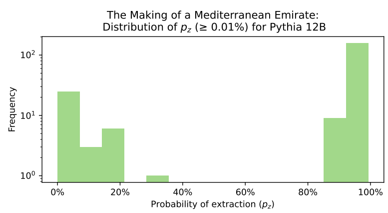
Figure 108: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.38. *The Making of a Mediterranean Emirate*, ROUIGHI

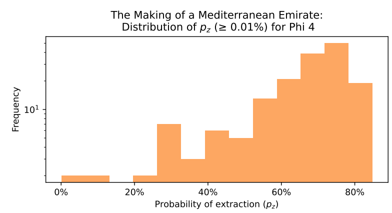


The Making of a Mediterranean Emirate				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.29%	0.34%	0.55%	00:18:22.98
PHI 4	0.28%	0.28%	1.06%	00:32:12.37
LLAMA 1 13B	0.30%	0.33%	1.73%	00:18:48.93
LLAMA 1 65B	0.30%	0.34%	2.14%	01:03:15.78
LLAMA 2 70B	0.30%	0.38%	2.53%	01:05:15.72
LLAMA 3 70B	0.28%	0.38%	4.44%	01:17:34.58
LLAMA 3.1 70B	0.28%	0.40%	4.74%	01:18:38.46
DEEPSEEK V1 67B	0.32%	0.43%	2.37%	01:29:44.07
QWEN 2.5 72B	0.30%	0.39%	2.22%	01:38:18.63
GEMMA 2 27B	0.30%	0.35%	1.12%	01:22:59.69

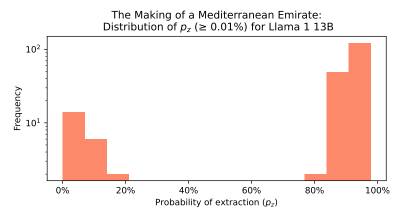
Figure 109: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



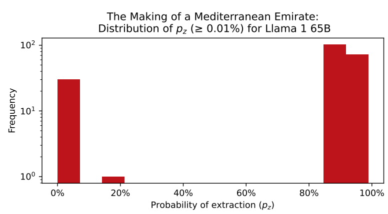
(a) PYTHIA 12B



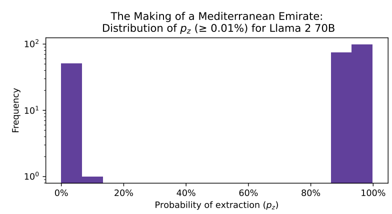
(b) PHI 4 (14B)



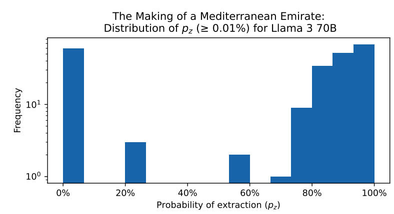
(c) LLAMA 1 13B



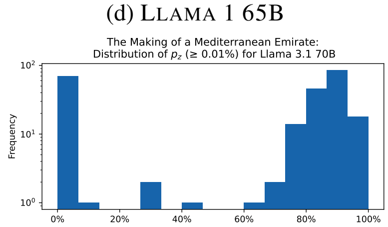
(d) LLAMA 1 65B



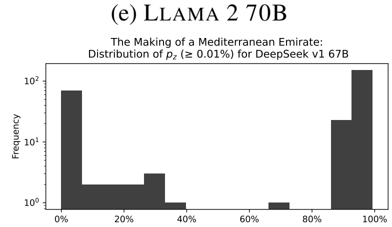
(e) LLAMA 2 70B



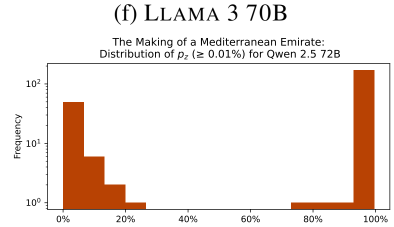
(f) LLAMA 3 70B



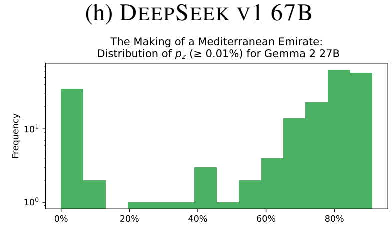
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



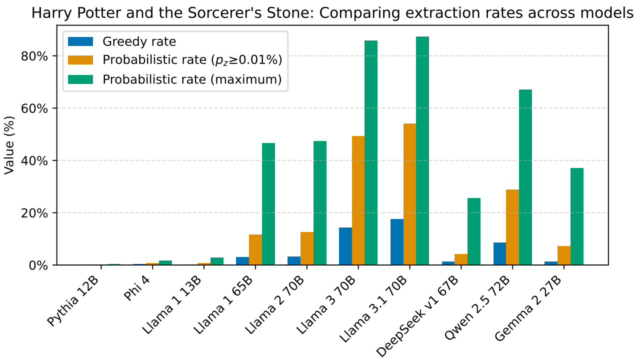
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

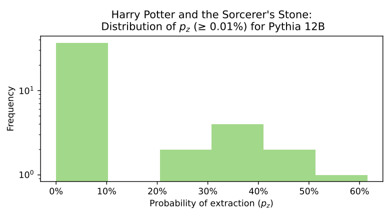
Figure 110: *The Making of a Mediterranean Emirate*, Rouighi (2011): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.39. *Harry Potter and the Sorcerer's Stone*, ROWLING

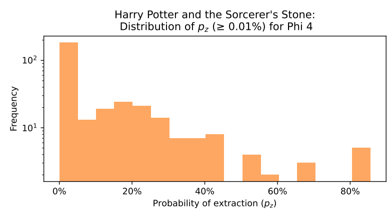


Harry Potter and the Sorcerer's Stone				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.04%	0.10%	0.39%	00:31:59.52
PHI 4	0.28%	0.67%	1.59%	00:57:48.55
LLAMA 1 13B	0.14%	0.60%	2.86%	00:20:10.69
LLAMA 1 65B	2.95%	11.51%	46.63%	00:56:26.10
LLAMA 2 70B	3.18%	12.47%	47.38%	00:49:32.78
LLAMA 3 70B	14.17%	49.24%	85.75%	01:00:39.76
LLAMA 3.1 70B	17.59%	54.08%	87.31%	01:02:48.59
DEEPSEEK V1 67B	1.19%	4.09%	25.58%	01:10:07.92
QWEN 2.5 72B	8.49%	28.86%	67.12%	01:17:14.85
GEMMA 2 27B	1.21%	7.11%	36.93%	01:05:32.95

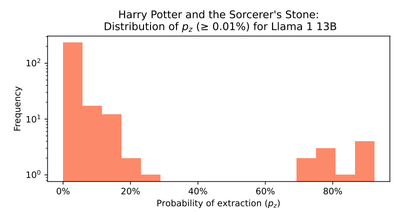
Figure 111: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



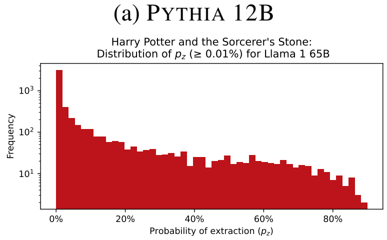
(a) PYTHIA 12B



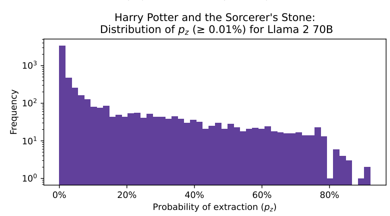
(b) PHI 4 (14B)



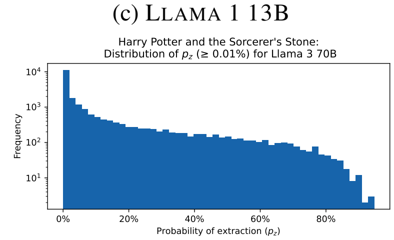
(c) LLAMA 1 13B



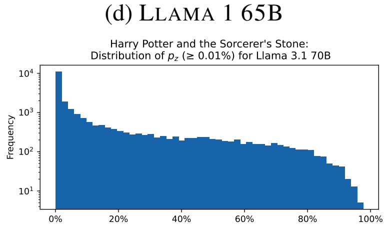
(d) LLAMA 1 65B



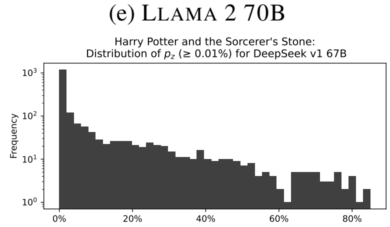
(e) LLAMA 2 70B



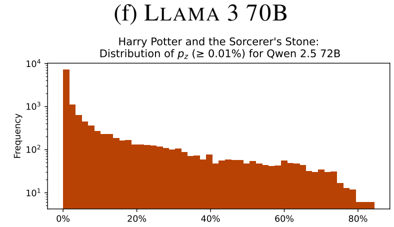
(f) LLAMA 3 70B



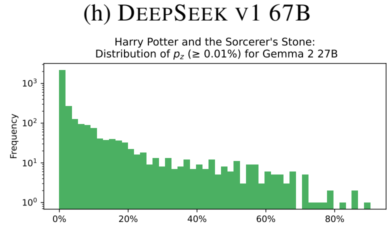
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 112: *Harry Potter and the Sorcerer's Stone*, Rowling (1998): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

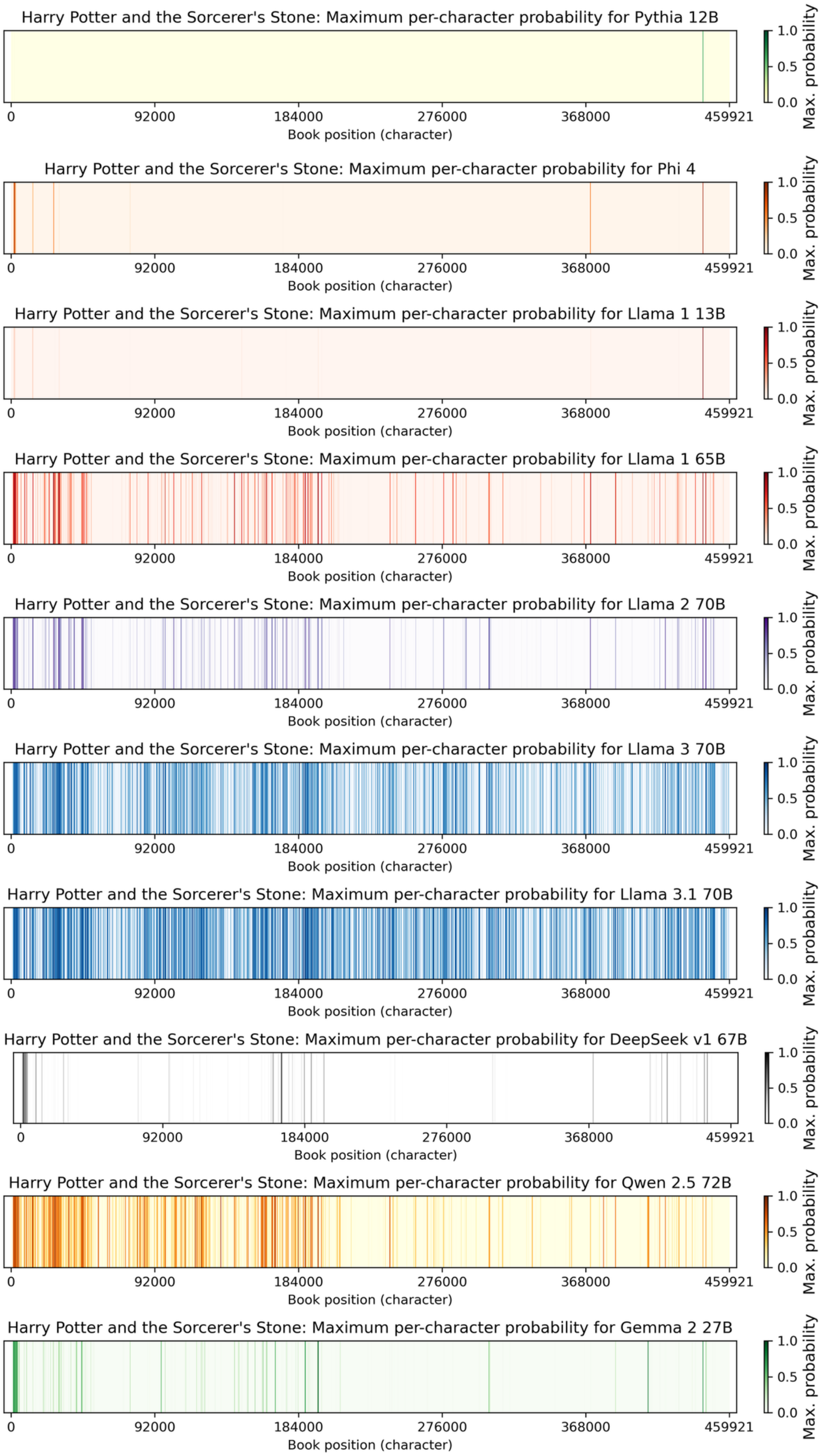
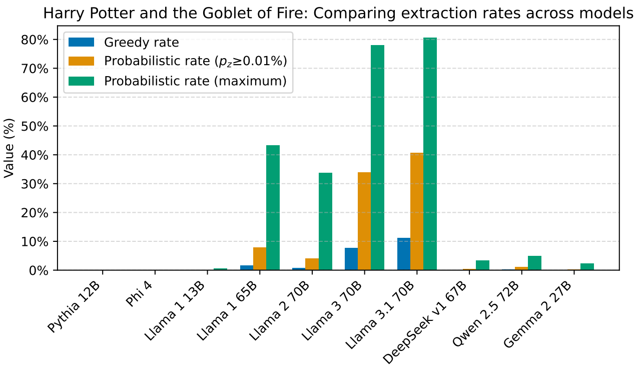


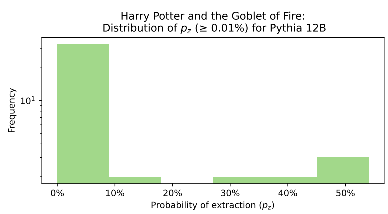
Figure 113: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.40. *Harry Potter and the Goblet of Fire*, ROWLING

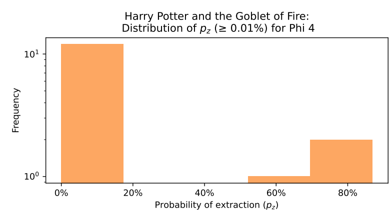


Harry Potter and the Goblet of Fire				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.01%	0.04%	0.17%	00:32:53.33
PHI 4	0.00%	0.01%	0.09%	01:02:10.06
LLAMA 1 13B	0.02%	0.12%	0.55%	00:35:20.49
LLAMA 1 65B	1.66%	7.82%	43.35%	02:03:02.22
LLAMA 2 70B	0.77%	4.12%	33.84%	02:01:51.02
LLAMA 3 70B	7.66%	33.96%	78.01%	02:27:34.90
LLAMA 3.1 70B	11.19%	40.71%	80.69%	02:33:44.93
DEEPSEEK V1 67B	0.16%	0.37%	3.37%	02:29:50.51
QWEN 2.5 72B	0.28%	1.20%	4.89%	02:39:47.84
GEMMA 2 27B	0.07%	0.35%	2.36%	02:20:58.57

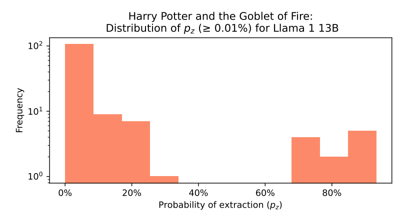
Figure 114: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



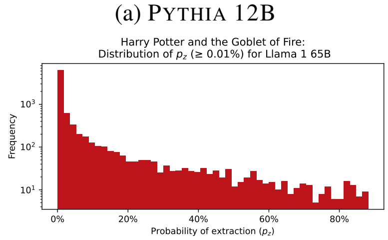
(a) PYTHIA 12B



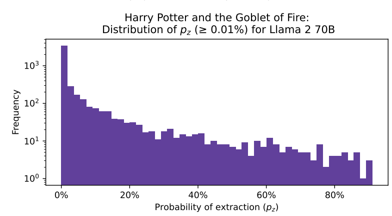
(b) PHI 4 (14B)



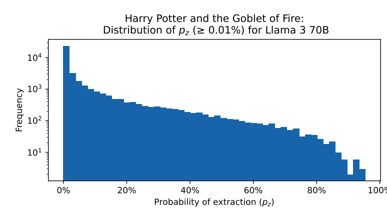
(c) LLAMA 1 13B



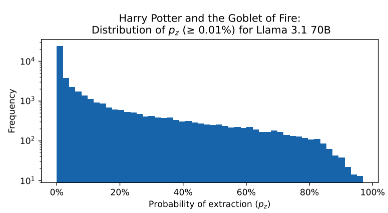
(d) LLAMA 1 65B



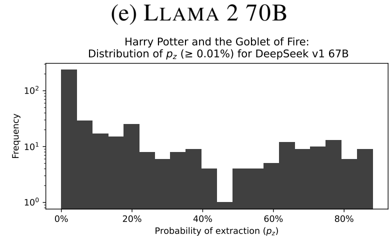
(e) LLAMA 2 70B



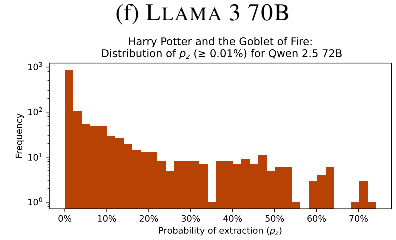
(f) LLAMA 3 70B



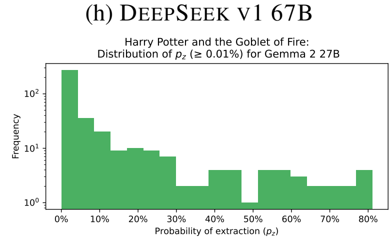
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 115: *Harry Potter and the Goblet of Fire*, Rowling (2000): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

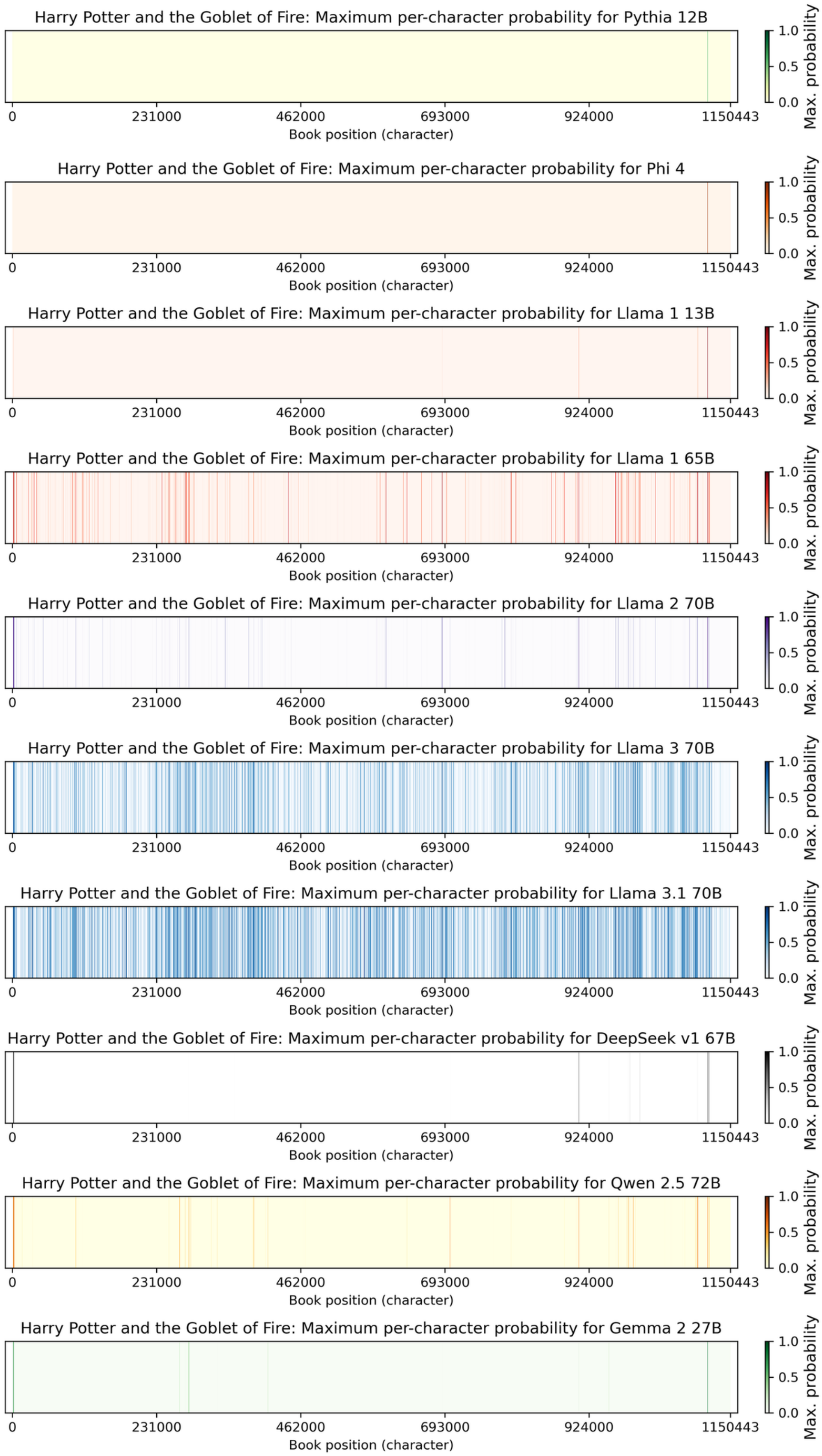
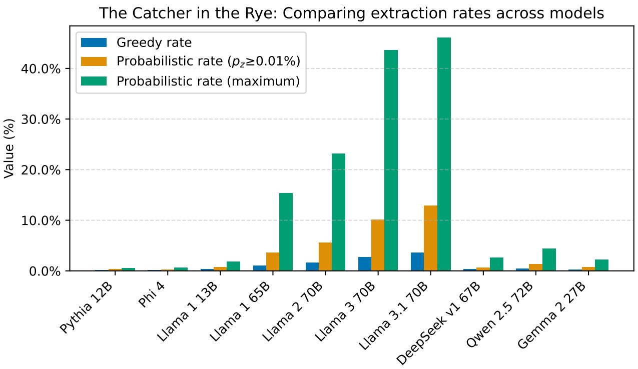


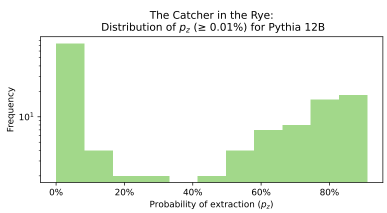
Figure 116: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.41. *The Catcher in the Rye*, SALINGER

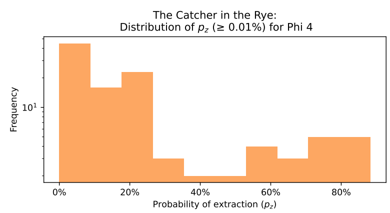


The Catcher in the Rye				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.17%	0.35%	0.49%	00:11:43.55
PHI 4	0.18%	0.28%	0.67%	00:20:24.79
LLAMA 1 13B	0.32%	0.72%	1.80%	00:12:00.49
LLAMA 1 65B	1.09%	3.56%	15.39%	00:41:42.65
LLAMA 2 70B	1.68%	5.60%	23.13%	00:42:18.63
LLAMA 3 70B	2.70%	10.09%	43.66%	00:50:52.85
LLAMA 3.1 70B	3.65%	12.91%	46.10%	00:51:32.40
DEEPSEEK V1 67B	0.36%	0.68%	2.64%	00:55:38.22
QWEN 2.5 72B	0.49%	1.30%	4.42%	00:55:06.67
GEMMA 2 27B	0.25%	0.69%	2.19%	00:46:51.43

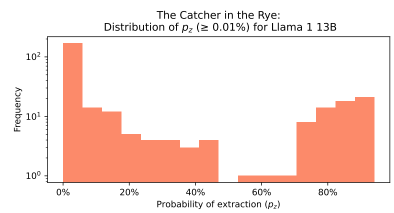
Figure 117: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



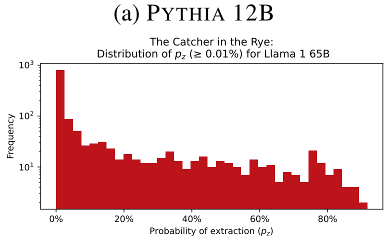
(a) PYTHIA 12B



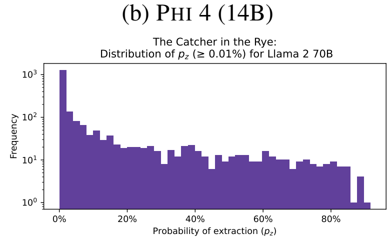
(b) PHI 4 (14B)



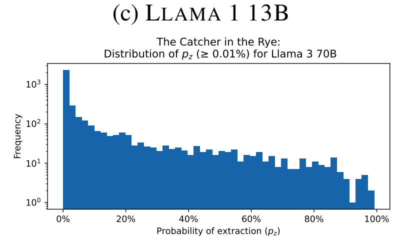
(c) LLAMA 1 13B



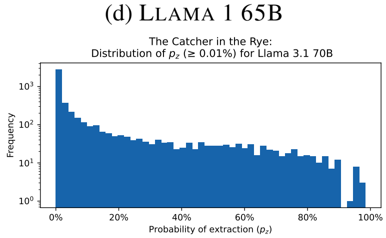
(d) LLAMA 1 65B



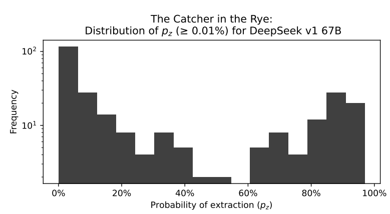
(e) LLAMA 2 70B



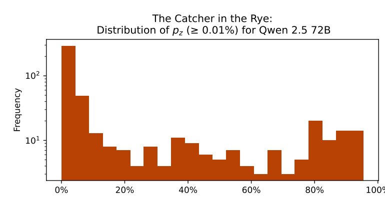
(f) LLAMA 3 70B



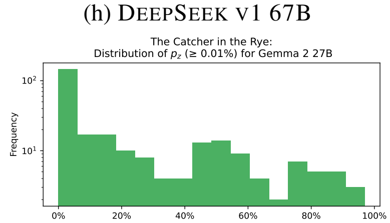
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 118: *The Catcher in the Rye*, Salinger (1951): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

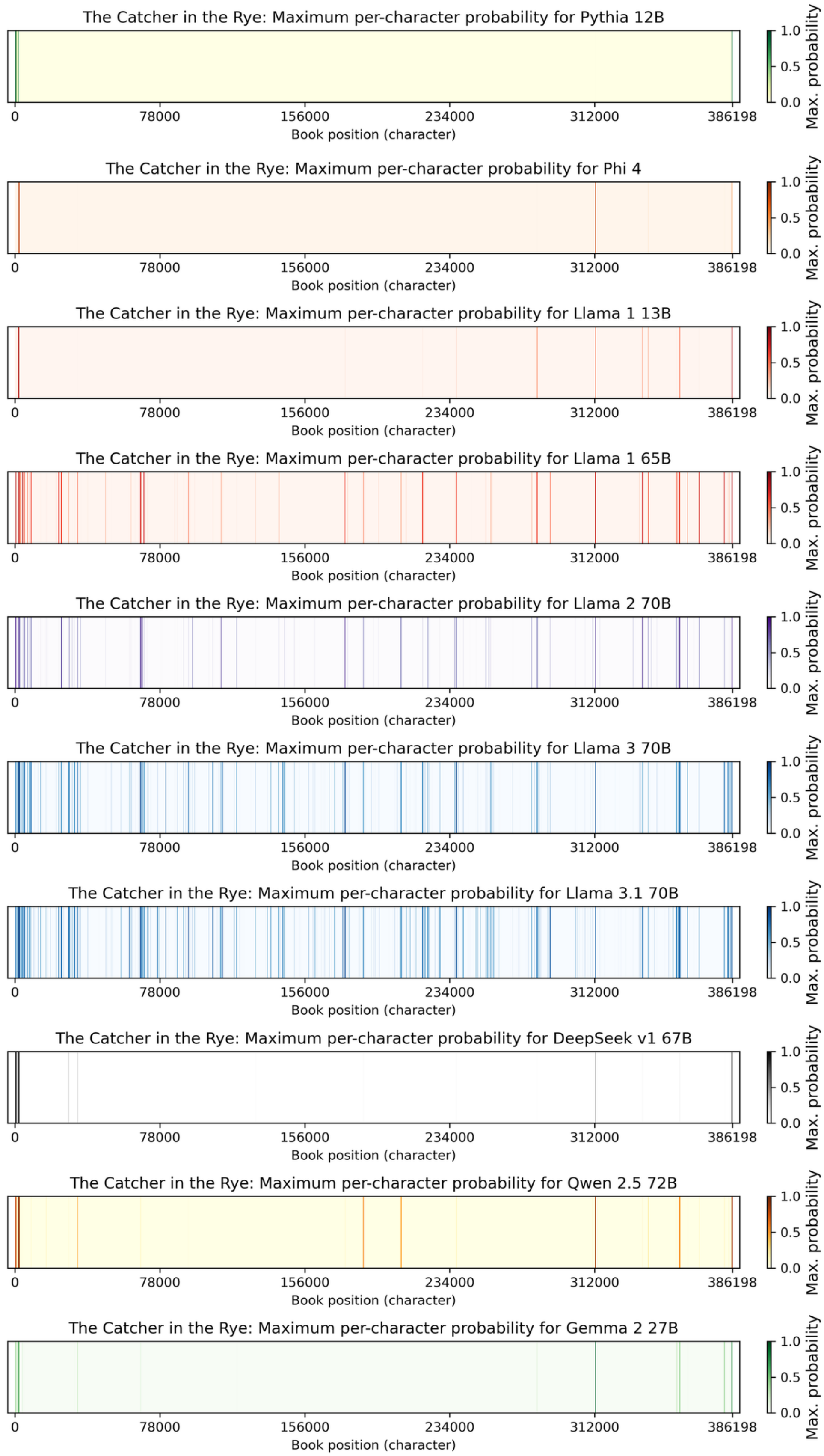
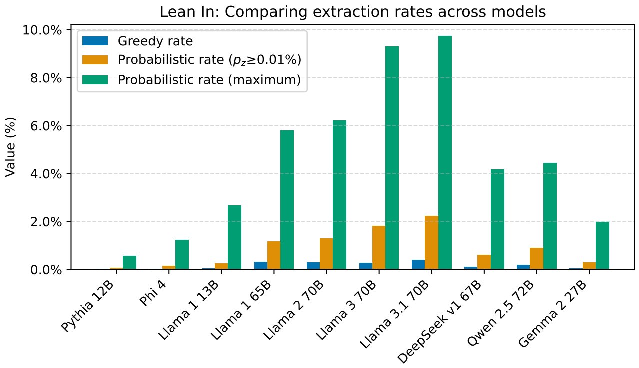


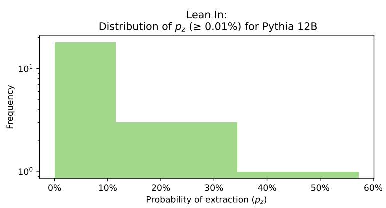
Figure 119: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.42. *Lean In*, SANDBERG

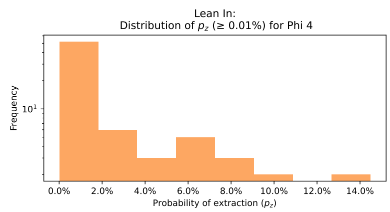


Lean In				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.03%	0.06%	0.56%	00:14:13.58
PHI 4	0.02%	0.16%	1.22%	00:26:31.67
LLAMA 1 13B	0.05%	0.25%	2.68%	00:15:00.22
LLAMA 1 65B	0.31%	1.18%	5.79%	00:49:39.22
LLAMA 2 70B	0.29%	1.28%	6.21%	00:50:18.34
LLAMA 3 70B	0.26%	1.81%	9.31%	01:02:16.57
LLAMA 3.1 70B	0.39%	2.23%	9.73%	01:02:39.76
DEEPSEEK V1 67B	0.11%	0.61%	4.16%	01:11:49.33
QWEN 2.5 72B	0.19%	0.91%	4.44%	01:19:04.88
GEMMA 2 27B	0.05%	0.29%	1.98%	01:05:46.30

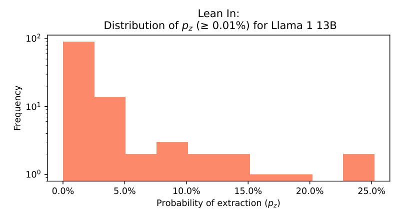
Figure 120: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



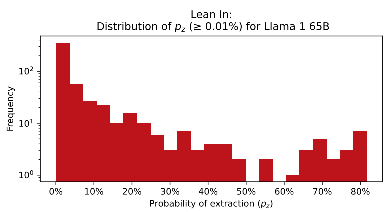
(a) PYTHIA 12B



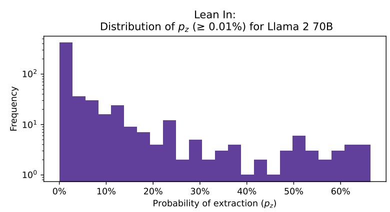
(b) PHI 4 (14B)



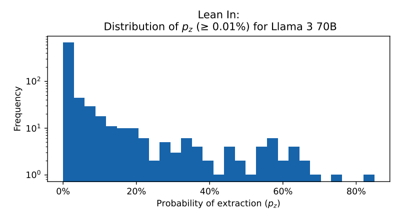
(c) LLAMA 1 13B



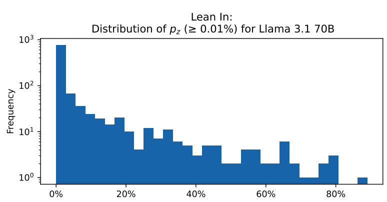
(d) LLAMA 1 65B



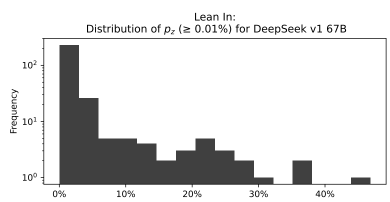
(e) LLAMA 2 70B



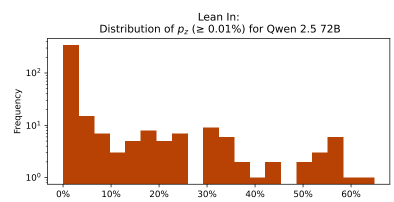
(f) LLAMA 3 70B



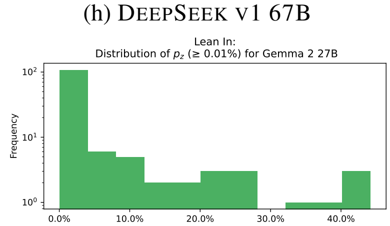
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 121: *Lean In*, Sandberg (2013): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models.

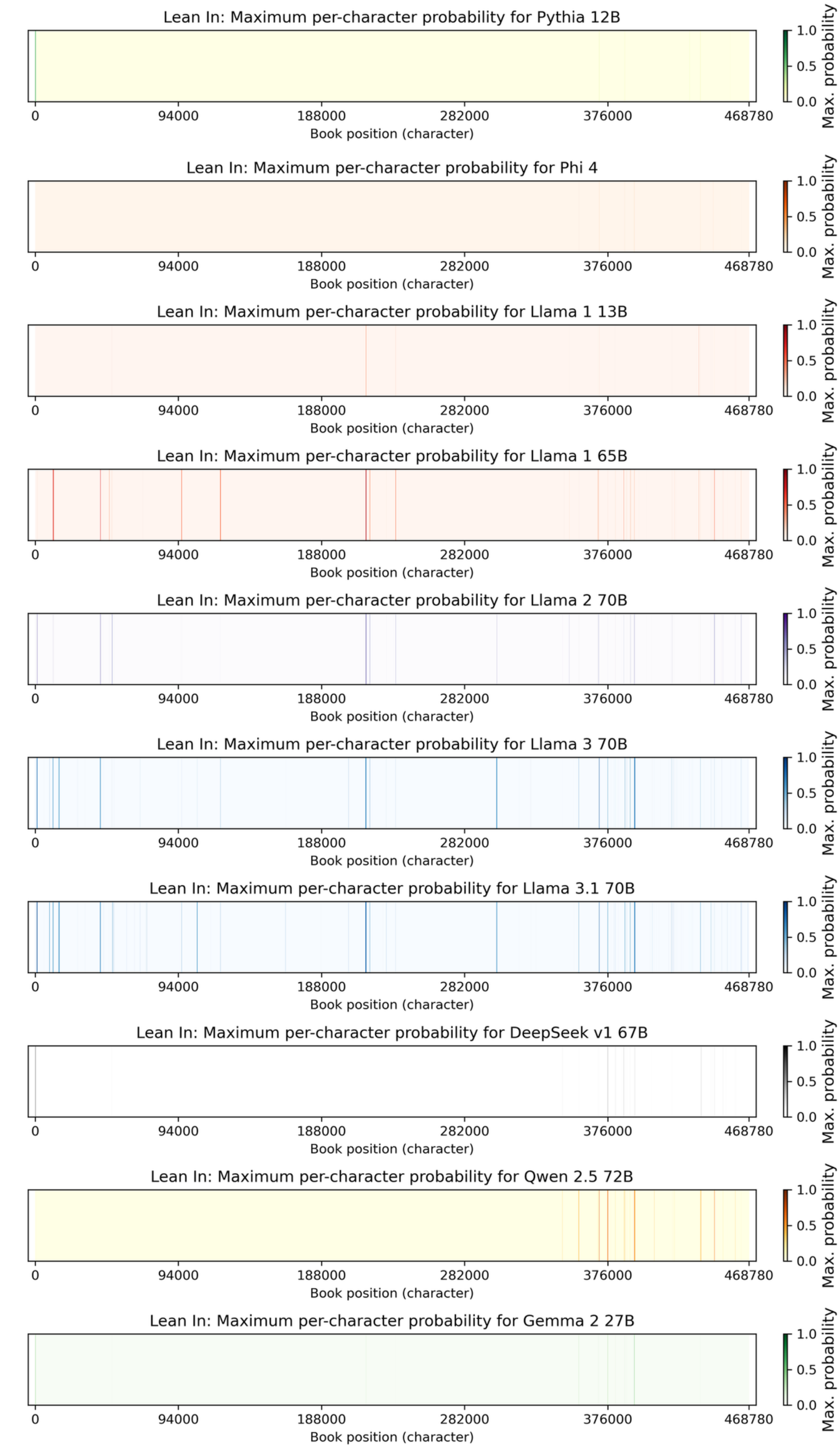
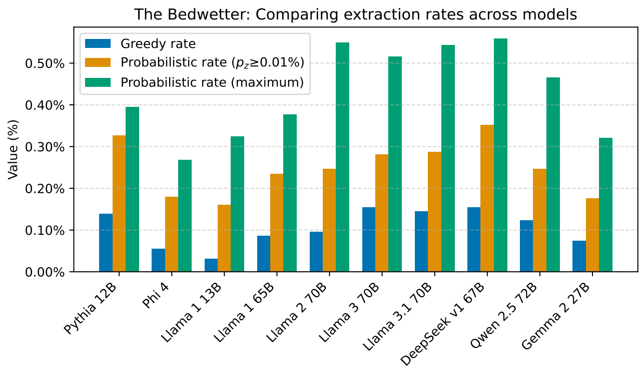


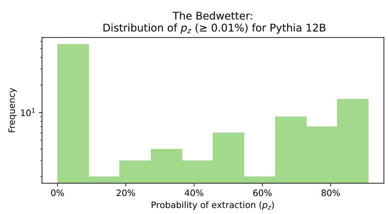
Figure 122: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.43. *The Bedwetter*, SILVERMAN

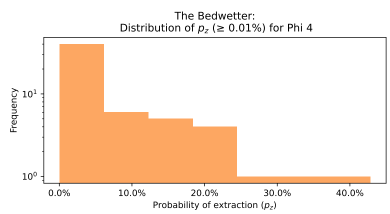


The Bedwetter				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.14%	0.33%	0.39%	00:21:15.89
PHI 4	0.06%	0.18%	0.27%	00:43:03.81
LLAMA 1 13B	0.03%	0.16%	0.32%	00:15:56.30
LLAMA 1 65B	0.09%	0.23%	0.38%	00:41:15.11
LLAMA 2 70B	0.10%	0.25%	0.55%	00:35:01.42
LLAMA 3 70B	0.15%	0.28%	0.52%	00:42:34.22
LLAMA 3.1 70B	0.15%	0.29%	0.54%	00:44:21.50
DEEPSEEK V1 67B	0.15%	0.35%	0.56%	00:52:18.98
QWEN 2.5 72B	0.12%	0.25%	0.47%	00:54:15.95
GEMMA 2 27B	0.07%	0.18%	0.32%	00:50:06.27

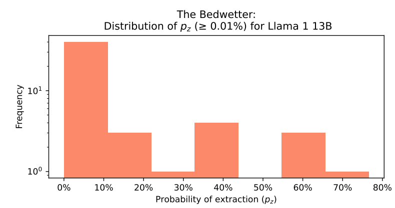
Figure 123: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



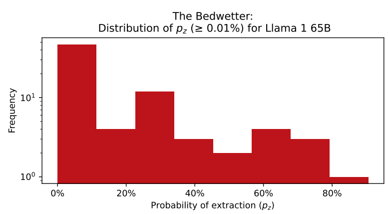
(a) PYTHIA 12B



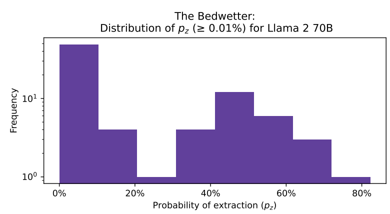
(b) PHI 4 (14B)



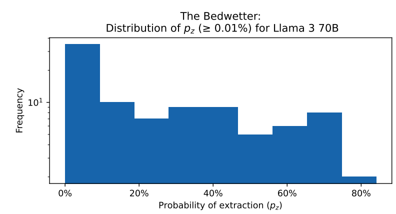
(c) LLAMA 1 13B



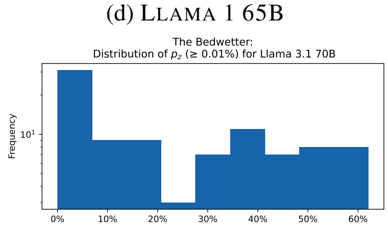
(d) LLAMA 1 65B



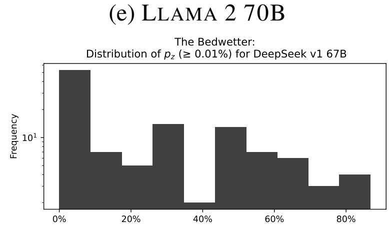
(e) LLAMA 2 70B



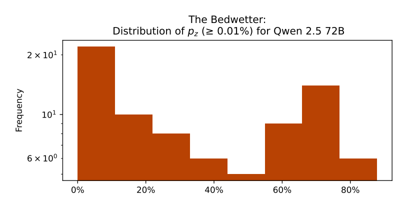
(f) LLAMA 3 70B



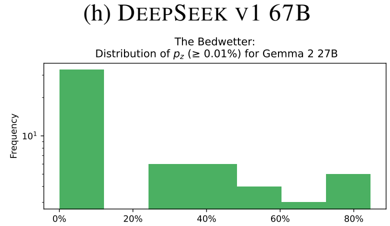
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



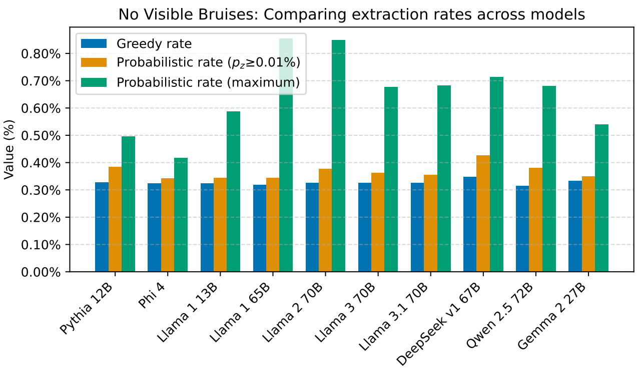
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

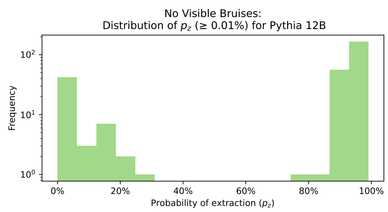
Figure 124: *The Bedwetter*, Silverman (2010): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

F.1.44. *No Visible Bruises*, SNYDER

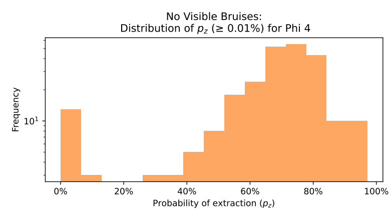


No Visible Bruises				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.33%	0.38%	0.50%	00:49:33.08
PHI 4	0.32%	0.34%	0.42%	01:36:16.77
LLAMA 1 13B	0.32%	0.34%	0.59%	00:33:52.32
LLAMA 1 65B	0.32%	0.34%	0.85%	01:30:24.99
LLAMA 2 70B	0.33%	0.38%	0.85%	01:17:15.88
LLAMA 3 70B	0.33%	0.36%	0.68%	01:33:06.62
LLAMA 3.1 70B	0.33%	0.36%	0.68%	01:35:40.82
DEEPSEEK V1 67B	0.35%	0.43%	0.71%	01:41:15.61
QWEN 2.5 72B	0.32%	0.38%	0.68%	01:55:37.64
GEMMA 2 27B	0.33%	0.35%	0.54%	01:37:50.88

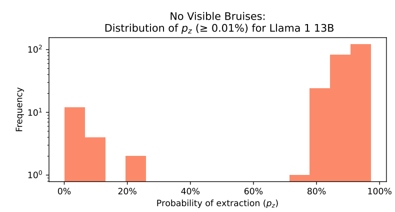
Figure 125: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_z \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



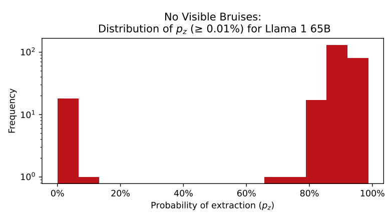
(a) PYTHIA 12B



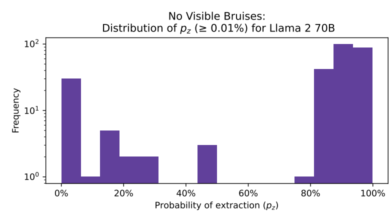
(b) PHI 4 (14B)



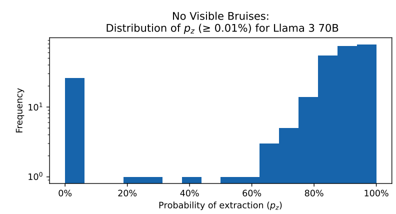
(c) LLAMA 1 13B



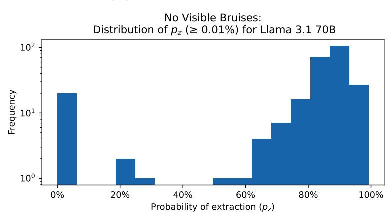
(d) LLAMA 1 65B



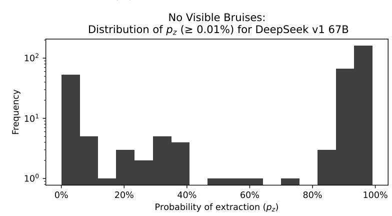
(e) LLAMA 2 70B



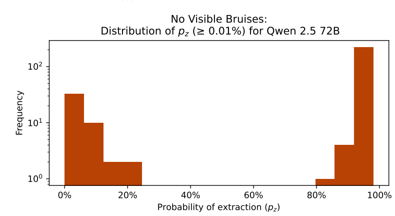
(f) LLAMA 3 70B



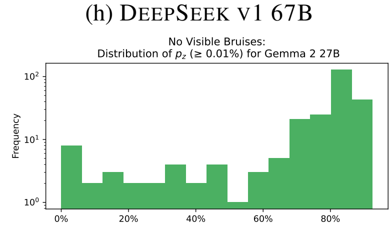
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



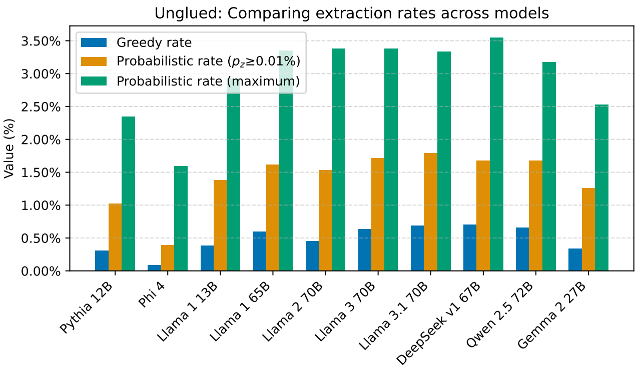
(i) QWEN 2.5 72B



(j) GEMMA 2 27B

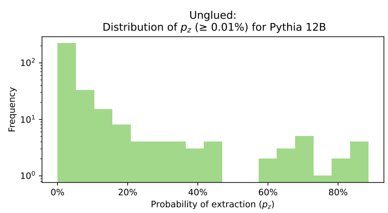
Figure 126: *No Visible Bruises*, Snyder (2019): Distributions over p_z (limited to $p_z \geq 0.01\%$) for different models.

F.1.45. *Unglued*, TERKEURST

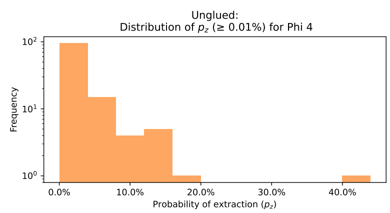


Unglued				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.31%	1.02%	2.35%	00:09:36.71
PHI 4	0.08%	0.39%	1.59%	00:18:33.90
LLAMA 1 13B	0.38%	1.38%	2.92%	00:09:39.71
LLAMA 1 65B	0.60%	1.62%	3.35%	00:33:11.48
LLAMA 2 70B	0.45%	1.53%	3.38%	00:33:34.66
LLAMA 3 70B	0.64%	1.71%	3.38%	00:40:36.62
LLAMA 3.1 70B	0.69%	1.79%	3.34%	00:40:17.82
DEEPSEEK V1 67B	0.70%	1.68%	3.55%	00:50:52.78
QWEN 2.5 72B	0.65%	1.67%	3.18%	00:59:44.79
GEMMA 2 27B	0.34%	1.26%	2.53%	00:49:42.29

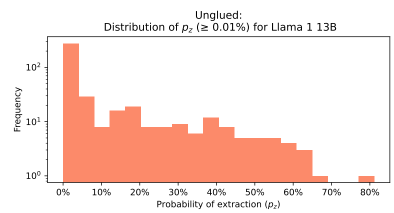
Figure 127: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



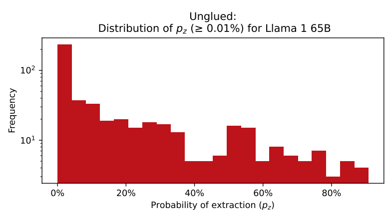
(a) PYTHIA 12B



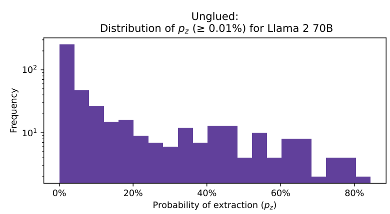
(b) PHI 4 (14B)



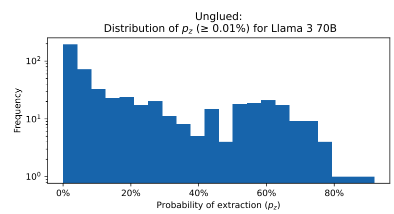
(c) LLAMA 1 13B



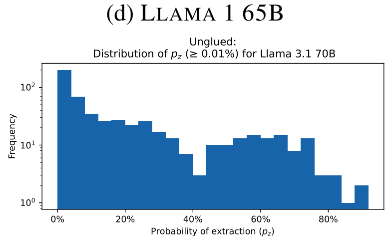
(d) LLAMA 1 65B



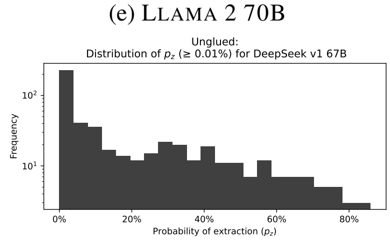
(e) LLAMA 2 70B



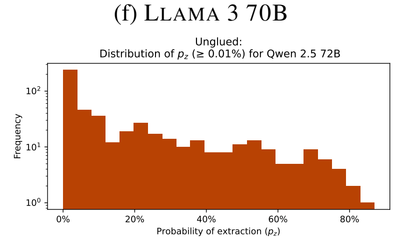
(f) LLAMA 3 70B



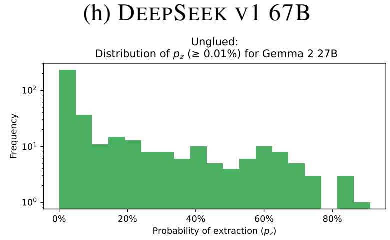
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 128: *Unglued*, TerKeurst (2012): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

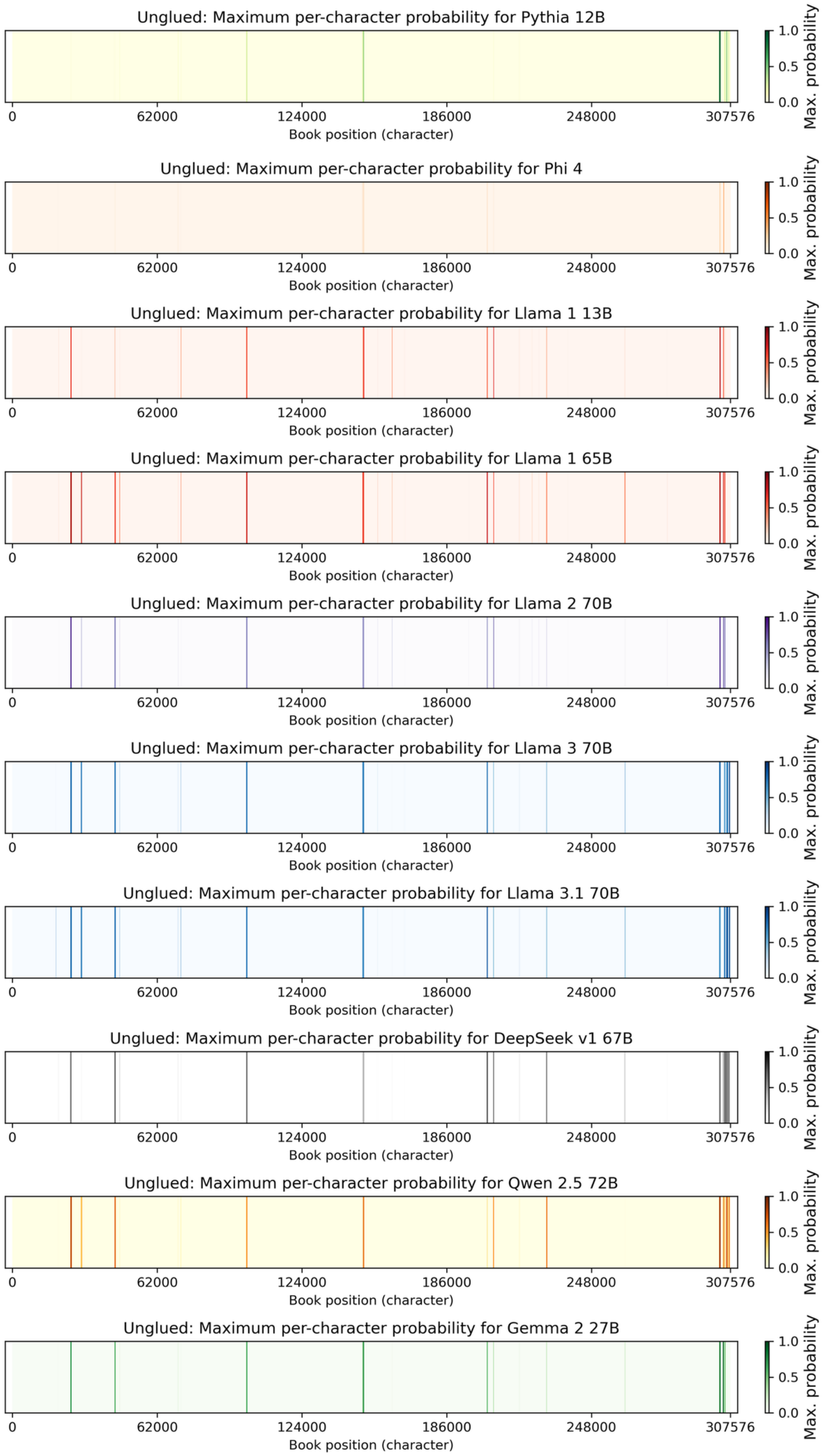
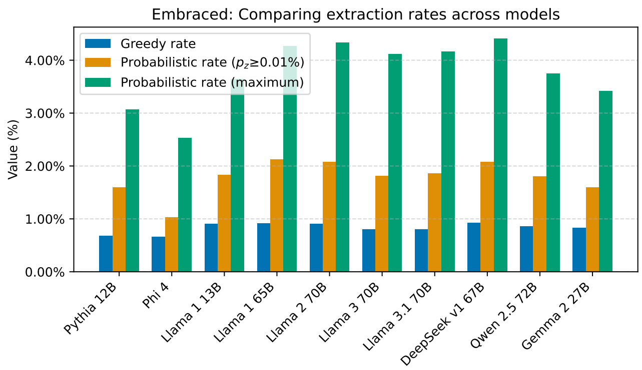


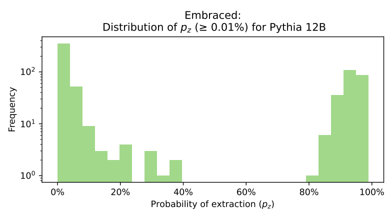
Figure 129: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.46. *Embraced*, TerKeurst

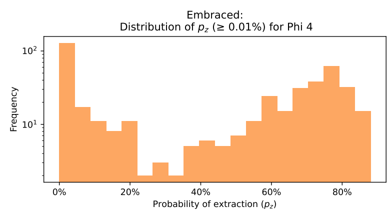


		Embraced		
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
Pythia 12B	0.68%	1.60%	3.07%	00:25:42.20
Phi 4	0.66%	1.03%	2.53%	00:49:09.94
Llama 1 13B	0.91%	1.83%	3.65%	00:19:19.73
Llama 1 65B	0.92%	2.13%	4.26%	00:52:07.03
Llama 2 70B	0.91%	2.08%	4.33%	00:45:19.79
Llama 3 70B	0.80%	1.81%	4.11%	00:54:43.37
Llama 3.1 70B	0.81%	1.86%	4.17%	00:57:18.49
DeepSeek v1 67B	0.93%	2.08%	4.41%	01:04:30.73
Qwen 2.5 72B	0.86%	1.80%	3.75%	01:11:41.35
Gemma 2 27B	0.83%	1.60%	3.42%	00:59:44.56

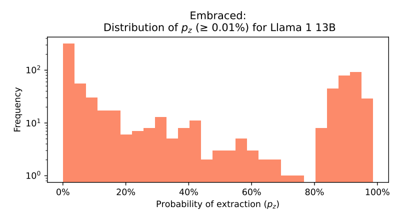
Figure 130: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



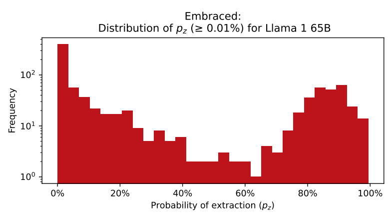
(a) Pythia 12B



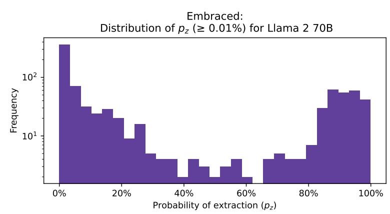
(b) Phi 4 (14B)



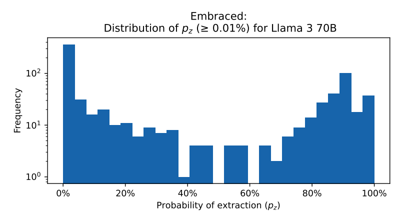
(c) Llama 1 13B



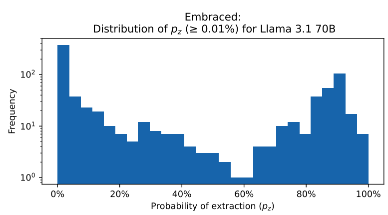
(d) Llama 1 65B



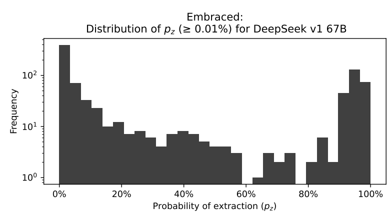
(e) Llama 2 70B



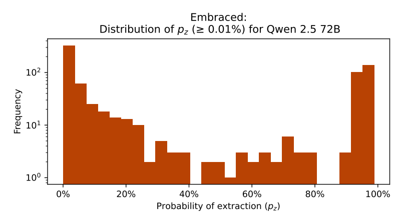
(f) Llama 3 70B



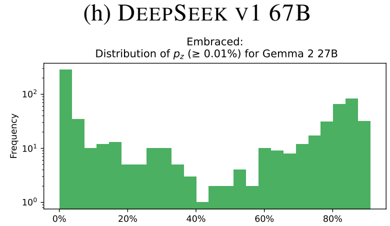
(g) Llama 3.1 70B



(h) DeepSeek v1 67B



(i) Qwen 2.5 72B



(j) Gemma 2 27B

Figure 131: *Embraced*, TerKeurst (2018): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

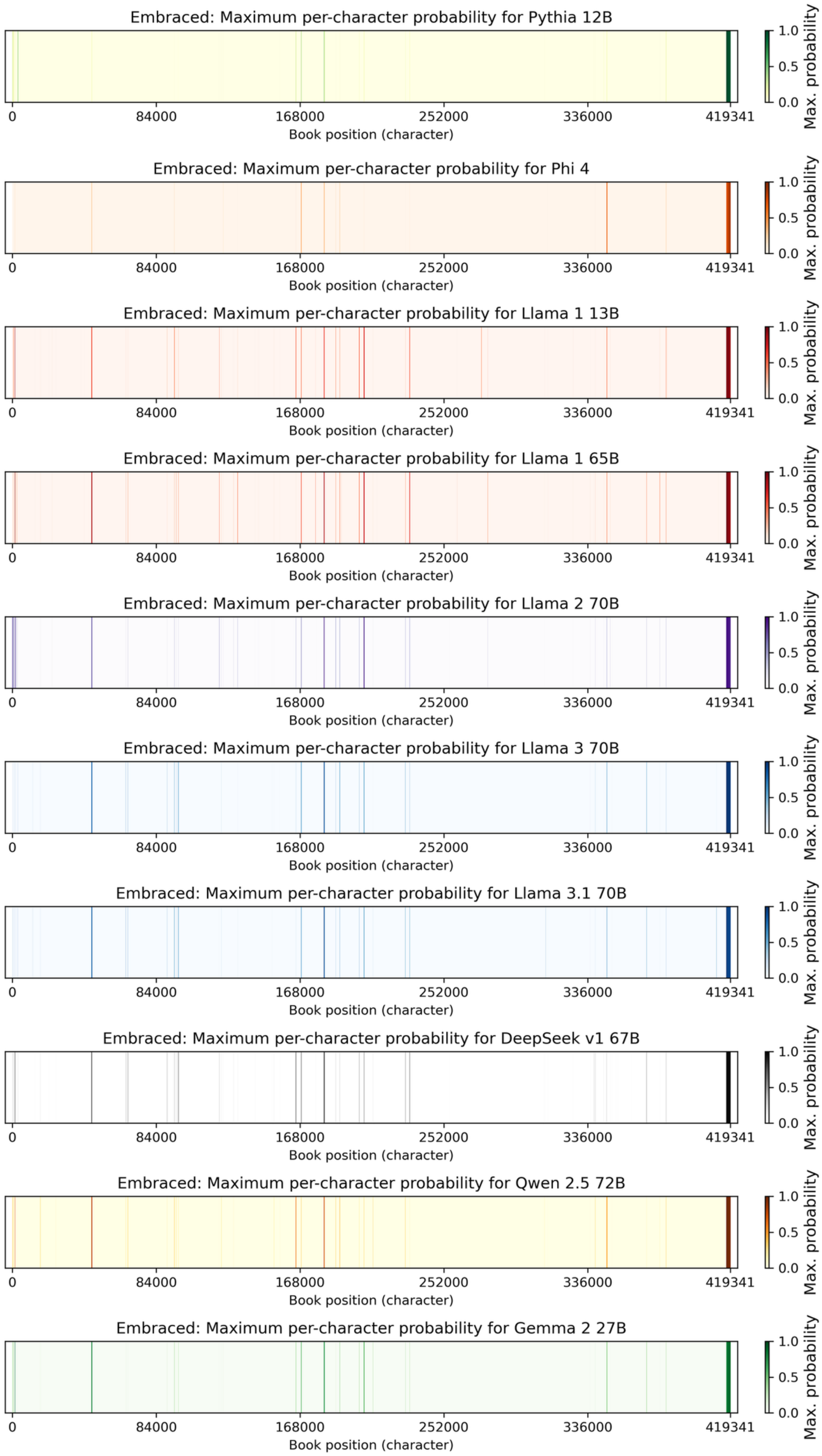
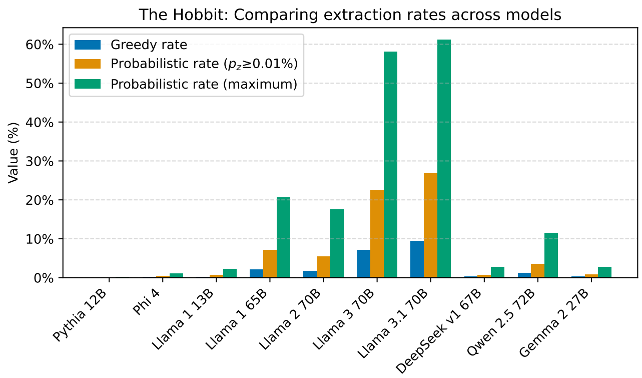


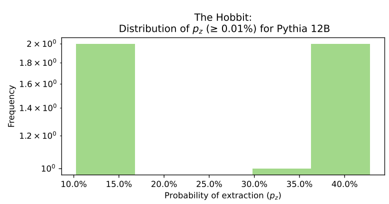
Figure 132: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.47. *The Hobbit*, TOLKIEN

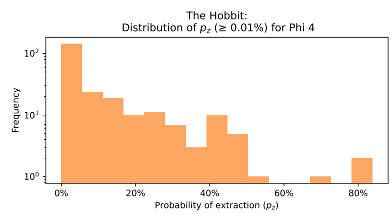


The Hobbit				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.01%	0.01%	0.21%	00:34:20.87
PHI 4	0.20%	0.46%	1.09%	01:08:05.60
LLAMA 1 13B	0.24%	0.74%	2.27%	00:25:50.33
LLAMA 1 65B	2.18%	7.10%	20.67%	01:05:58.13
LLAMA 2 70B	1.77%	5.52%	17.59%	00:55:37.34
LLAMA 3 70B	7.08%	22.63%	58.10%	01:06:14.12
LLAMA 3.1 70B	9.40%	26.88%	61.18%	01:07:54.81
DEEPSEEK V1 67B	0.33%	0.76%	2.81%	01:16:25.88
QWEN 2.5 72B	1.27%	3.52%	11.56%	01:20:55.39
GEMMA 2 27B	0.28%	0.89%	2.75%	01:14:35.10

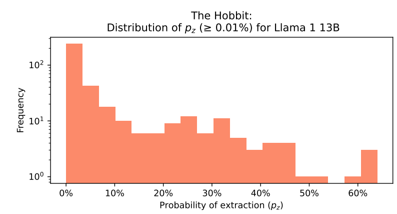
Figure 133: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



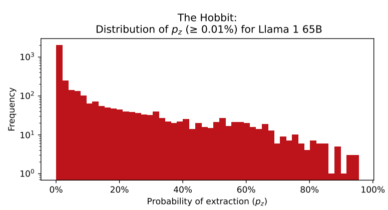
(a) PYTHIA 12B



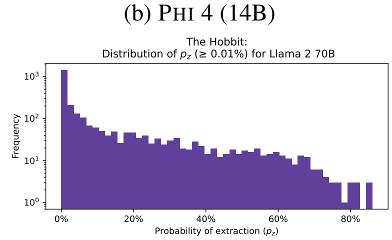
(b) PHI 4 (14B)



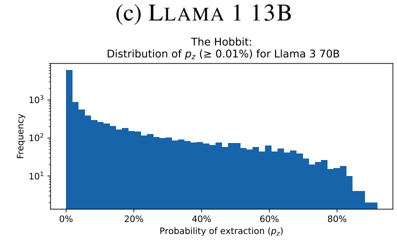
(c) LLAMA 1 13B



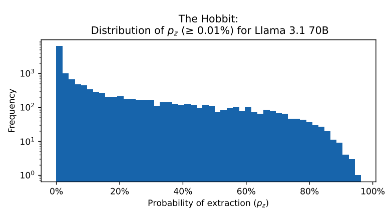
(d) LLAMA 1 65B



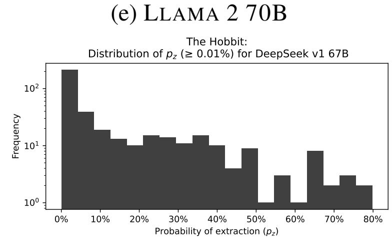
(e) LLAMA 2 70B



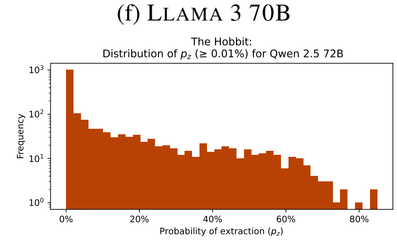
(f) LLAMA 3 70B



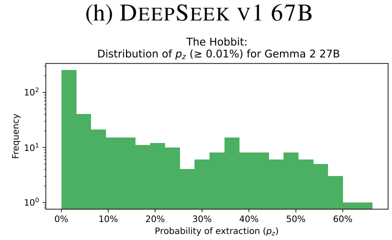
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 134: *The Hobbit*, Tolkien (1937): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

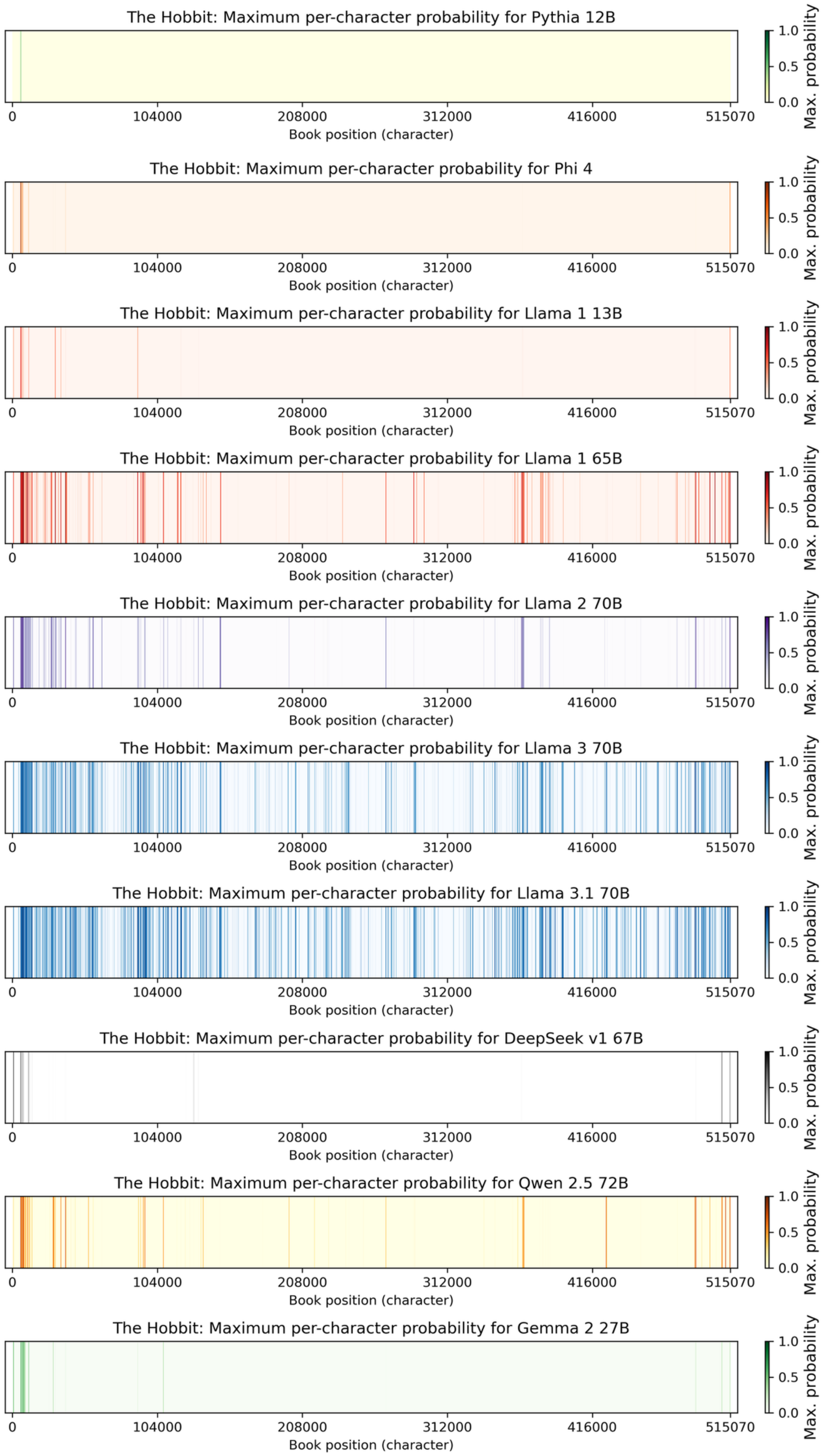
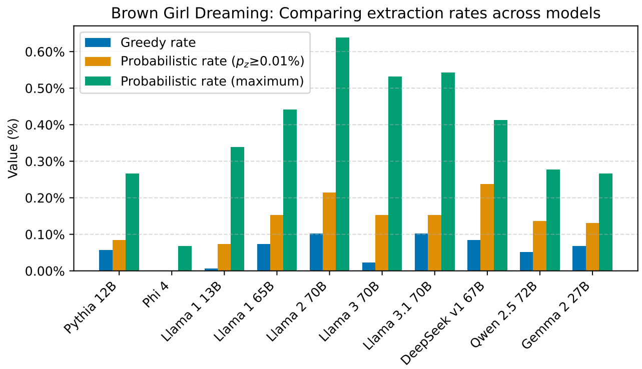


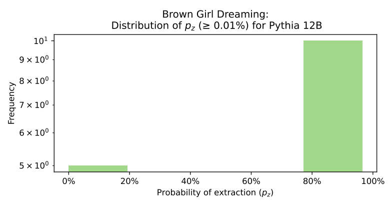
Figure 135: Heatmaps showing regions of extraction “hot-spots” according to location.

F.1.48. *Brown Girl Dreaming*, WOODSON

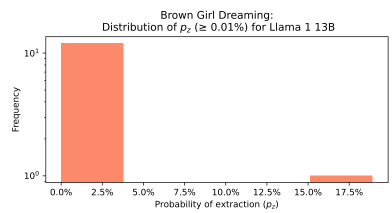


Brown Girl Dreaming				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.06%	0.08%	0.27%	00:10:39.32
PHI 4	0.00%	0.00%	0.07%	00:23:42.67
LLAMA 1 13B	0.01%	0.07%	0.34%	00:08:01.68
LLAMA 1 65B	0.07%	0.15%	0.44%	00:22:08.44
LLAMA 2 70B	0.10%	0.21%	0.64%	00:19:40.27
LLAMA 3 70B	0.02%	0.15%	0.53%	00:24:06.87
LLAMA 3.1 70B	0.10%	0.15%	0.54%	00:24:12.94
DEEPSEEK V1 67B	0.08%	0.24%	0.41%	00:33:58.72
QWEN 2.5 72B	0.05%	0.14%	0.28%	00:39:30.65
GEMMA 2 27B	0.07%	0.13%	0.27%	00:31:24.91

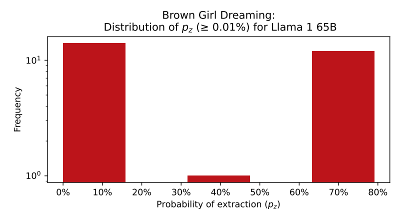
Figure 136: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



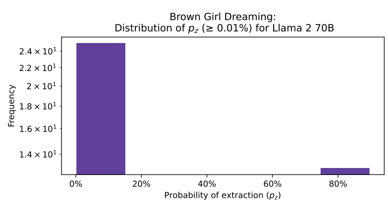
(a) PYTHIA 12B



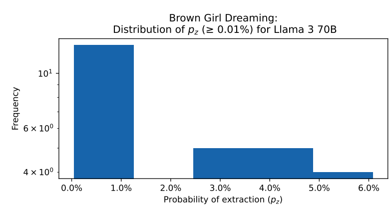
(b) LLAMA 1 13B



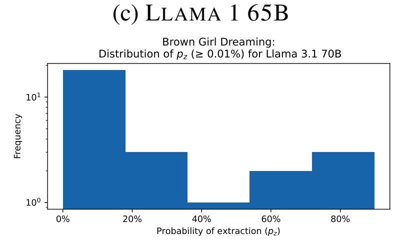
(c) LLAMA 1 65B



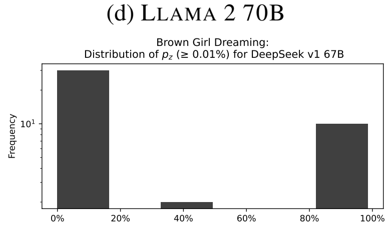
(d) LLAMA 2 70B



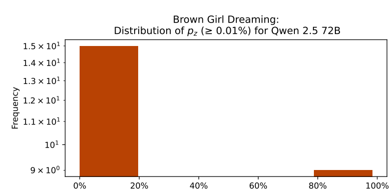
(e) LLAMA 3 70B



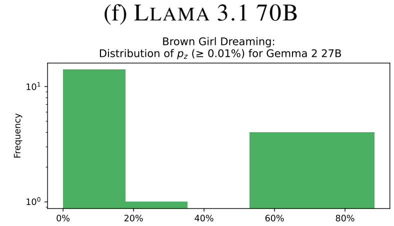
(f) LLAMA 3.1 70B



(g) DEEPSEEK V1 67B



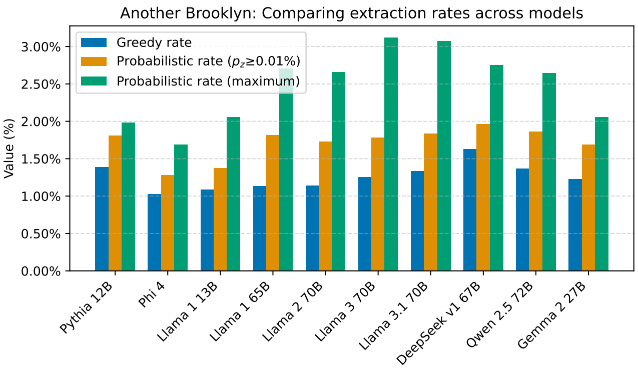
(h) QWEN 2.5 72B



(i) GEMMA 2 27B

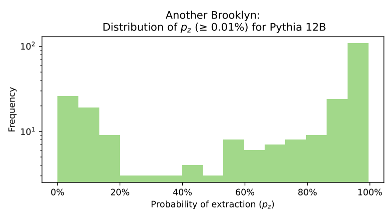
Figure 137: *Brown Girl Dreaming*, Woodson (2014): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models. Only showing models for which there exist any $p_{\boldsymbol{z}} \geq 0.01\%$.

F.1.49. *Another Brooklyn*, WOODSON

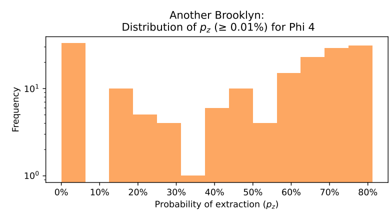


Another Brooklyn				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	1.39%	1.81%	1.98%	00:04:24.36
PHI 4	1.03%	1.28%	1.69%	00:09:55.09
LLAMA 1 13B	1.09%	1.37%	2.05%	00:04:19.46
LLAMA 1 65B	1.13%	1.81%	2.71%	00:15:29.82
LLAMA 2 70B	1.14%	1.73%	2.66%	00:14:57.47
LLAMA 3 70B	1.25%	1.79%	3.12%	00:18:22.40
LLAMA 3.1 70B	1.34%	1.84%	3.08%	00:18:25.37
DEEPSEEK V1 67B	1.63%	1.96%	2.75%	00:29:30.56
QWEN 2.5 72B	1.36%	1.86%	2.65%	00:33:11.13
GEMMA 2 27B	1.23%	1.69%	2.05%	00:26:43.76

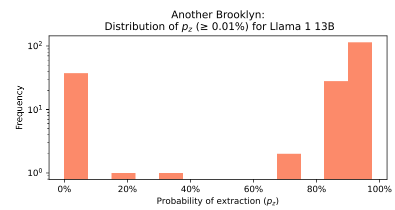
Figure 138: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



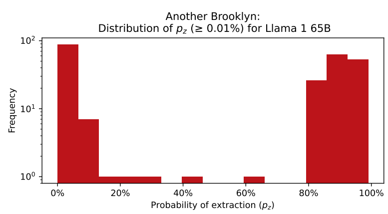
(a) PYTHIA 12B



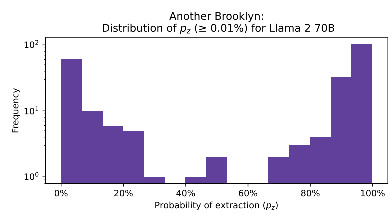
(b) PHI 4 (14B)



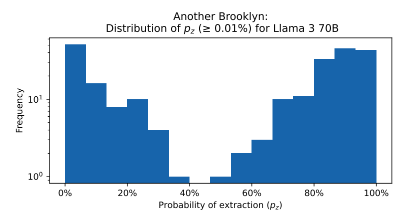
(c) LLAMA 1 13B



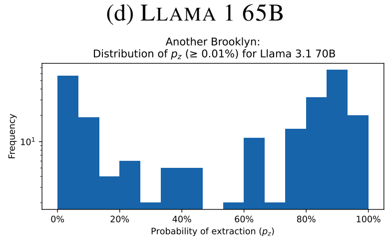
(d) LLAMA 1 65B



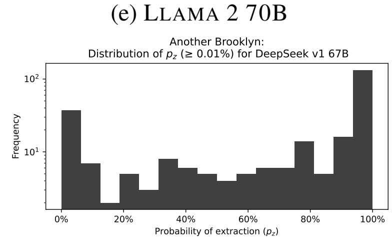
(e) LLAMA 2 70B



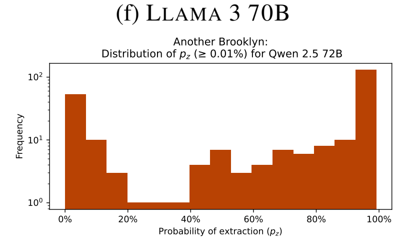
(f) LLAMA 3 70B



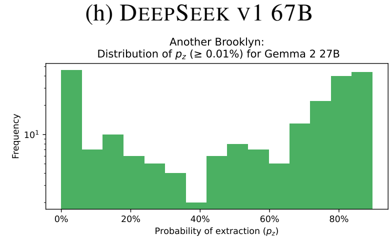
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 139: *Another Brooklyn*, Woodson (2016): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

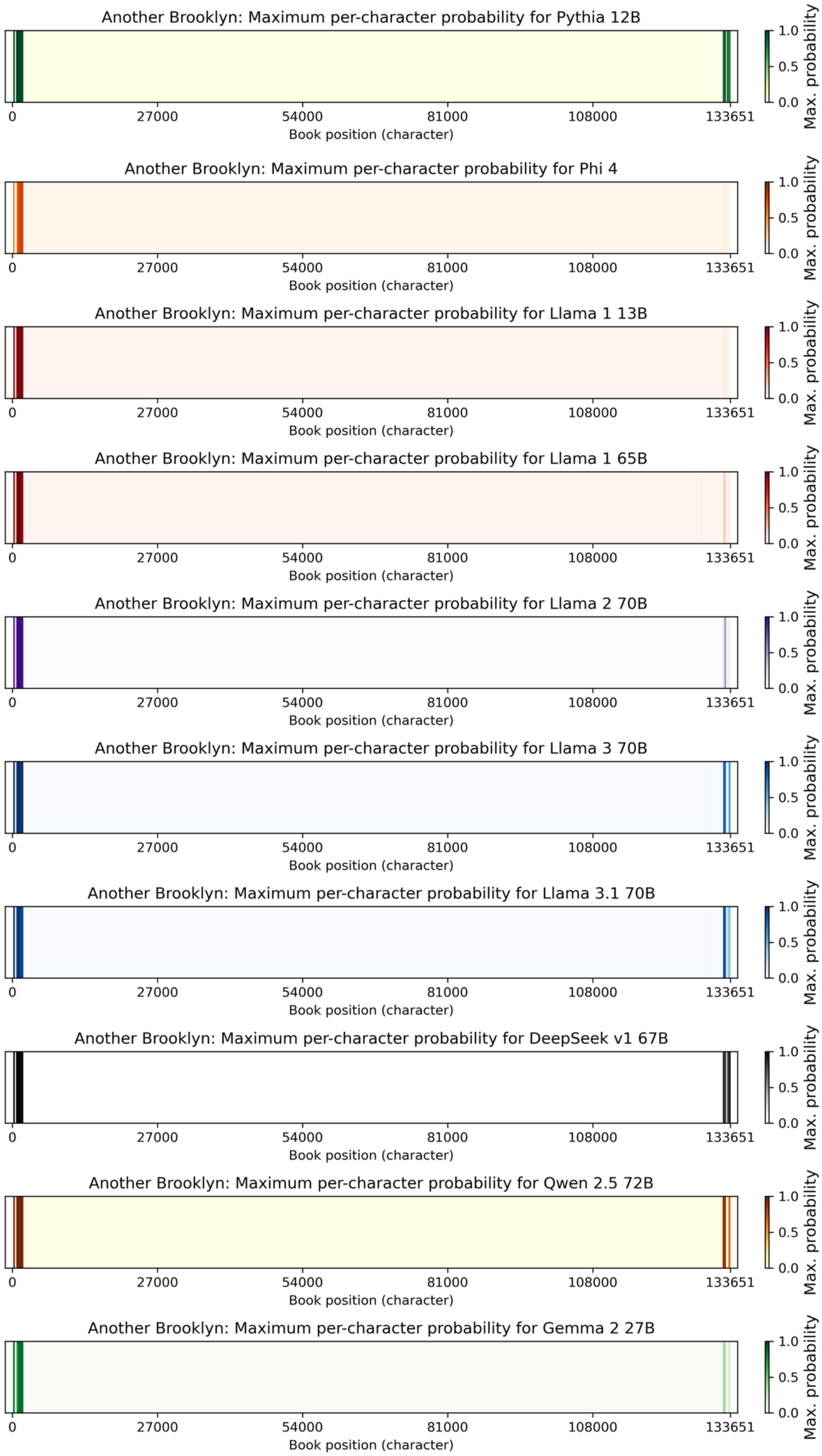
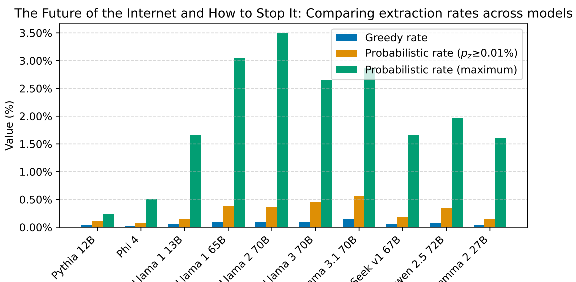


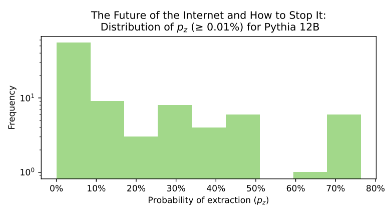
Figure 140: Heatmaps showing regions of extraction "hot-spots" according to location.

F.1.50. *The Future of the Internet and How to Stop It*, ZITTRAIN

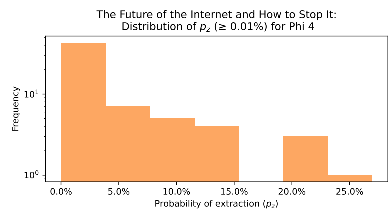


The Future of the Internet and How to Stop It				
Model	**Greedy**	$p_z \geq 0.01\%$	**Max.**	**Runtime (hh:mm:ss)**
PYTHIA 12B	0.04%	0.10%	0.23%	00:58:48.11
PHI 4	0.02%	0.07%	0.50%	01:53:41.34
LLAMA 1 13B	0.05%	0.15%	1.66%	00:42:27.92
LLAMA 1 65B	0.09%	0.38%	3.04%	01:51:51.40
LLAMA 2 70B	0.09%	0.36%	3.49%	01:37:27.03
LLAMA 3 70B	0.10%	0.46%	2.64%	01:58:01.62
LLAMA 3.1 70B	0.14%	0.57%	2.86%	02:00:02.33
DEEPSEEK V1 67B	0.06%	0.18%	1.66%	02:02:38.44
QWEN 2.5 72B	0.07%	0.35%	1.96%	02:19:50.42
GEMMA 2 27B	0.04%	0.15%	1.60%	02:05:30.57

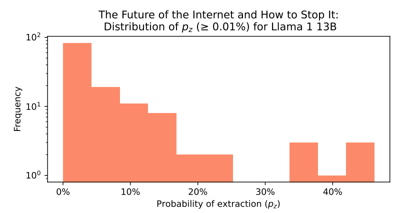
Figure 141: Comparing greedy discoverable extraction rate, probabilistic extraction rate for a conservative setting ($p_{\boldsymbol{z}} \geq 0.01\%$), and maximum possible (n, p)-discoverable rate across different models.



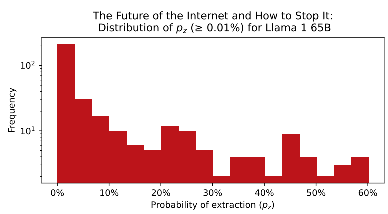
(a) PYTHIA 12B



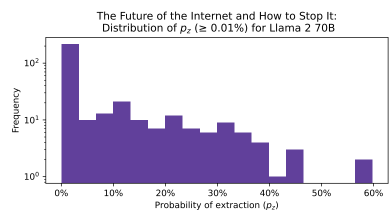
(b) PHI 4 (14B)



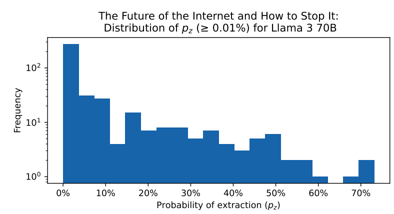
(c) LLAMA 1 13B



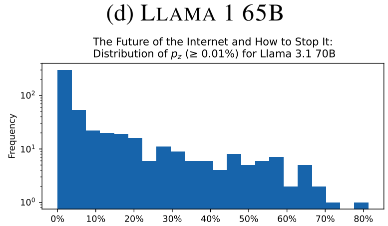
(d) LLAMA 1 65B



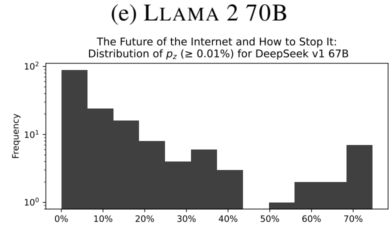
(e) LLAMA 2 70B



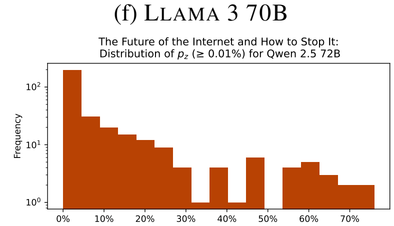
(f) LLAMA 3 70B



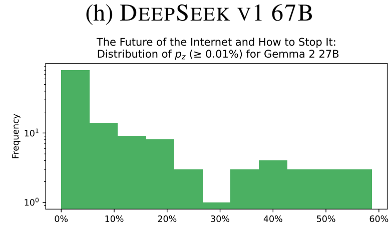
(g) LLAMA 3.1 70B



(h) DEEPSEEK V1 67B



(i) QWEN 2.5 72B



(j) GEMMA 2 27B

Figure 142: *The Future of the Internet and How to Stop It*, Zittrain (2008): Distributions over $p_{\boldsymbol{z}}$ (limited to $p_{\boldsymbol{z}} \geq 0.01\%$) for different models.

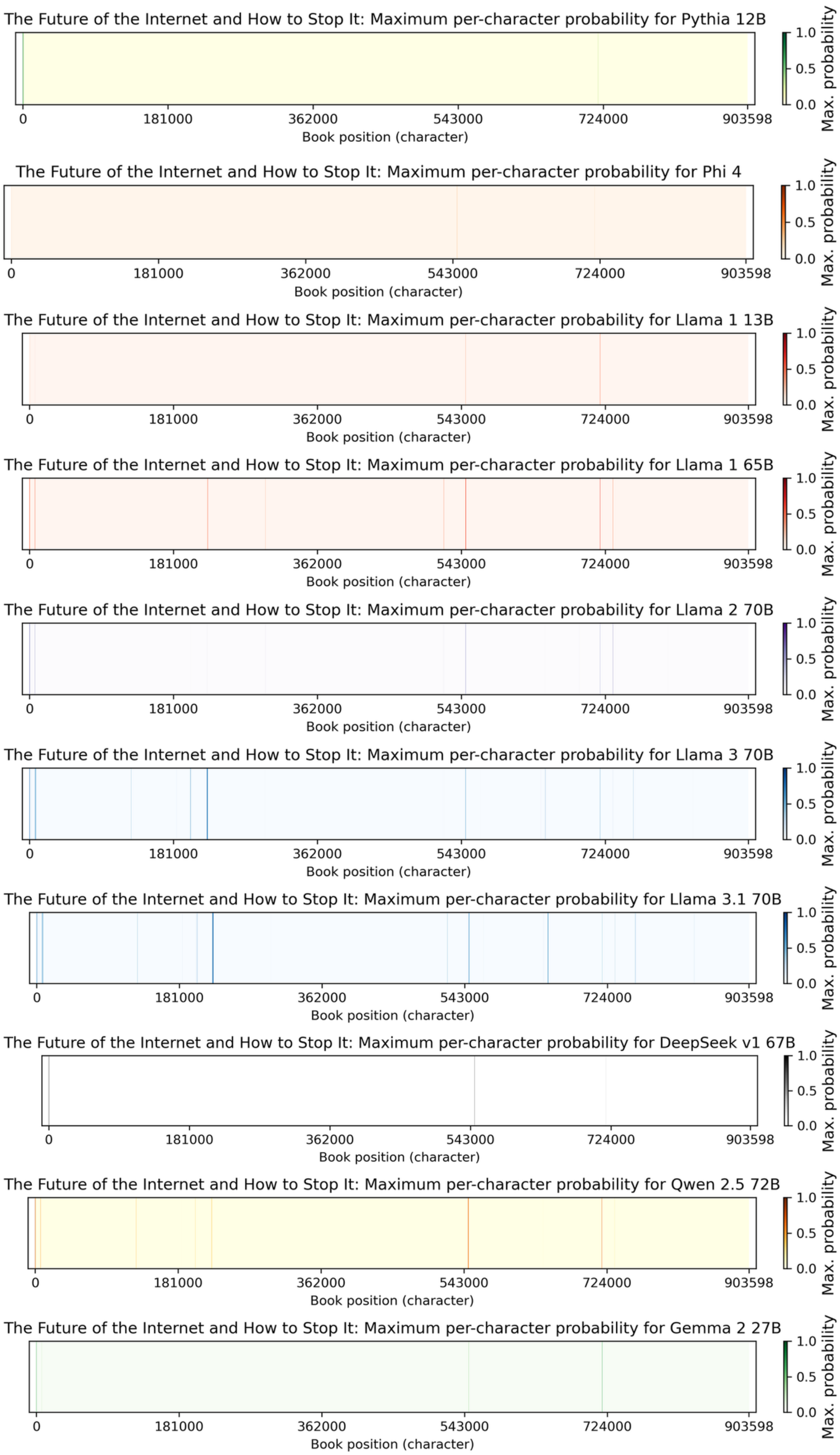


Figure 143: Heatmaps showing regions of extraction “hot-spots” according to location.

G. Selection of extracted examples

This project generated tens of thousands of extracted examples. We could not evaluate them all manually. We provide a brief selection of arbitrarily chosen extracted examples from some of the 36 books in the sliding window experiments. (See Section 4 & Appendix F.1). We defer discussion of these examples (and our book-level results) to Appendix H. We also refer to some of these examples in Section 4. Unless otherwise specified, all examples have a prefix length of 50 tokens and a suffix length of 50 tokens.

Almost all books indicate high-probability extraction for the copyright notice. This is highly duplicated text, so it is not necessarily the case that the high extraction probability/ memorization is *due to* the exact text from which we extract it. It is still memorized, under this definition of memorization. The relationship between duplication and memorization is well-documented (Lee et al., 2022). Similarly, publisher addresses, authors biographies (which are also highly duplicated text on the internet) also show up as extractable for many books. We provide some non-exhaustive examples of this for some books.

Another Brooklyn **(Woodson, 2016)**

LLAMA 3.1 70B

(Copyright notice)

Prefix: ‘RIGHT\n\n ANOTHER BROOKLYN. Copyright © 2016 by Jacqueline Woodson. All rights reserved under International and Pan-American Copyright Conventions. By payment of the required fees, you have been granted the nonexclusive, nontransferable’

Suffix: ‘ right to access and read the text of this e-book on-screen. No part of this text may be reproduced, transmitted, downloaded, decompiled, reverse-engineered, or stored in or introduced into any information storage and retrieval system, in any form’

Sequence probability ($p_{\boldsymbol{z}}$): 0.825882886427520

(Publisher address)

Prefix: ‘treet\n\nSydney, NSW 2000, Australia\n\nwww.harpercollins.com.au\n\n**Canada**\n\n HarperCollins Canada\n\n2 Bloor Street East - 20th Floor\n\nToronto, ON M4W 1’

Suffix: ‘ A8, Canada\n\nwww.harpercollins.ca\n\n**New Zealand**\n\nHarperCollins Publishers New Zealand\n\nUnit D1, 63 Apollo Drive\n\nRosedale 0632\n\nAuckland, New Zealand\n\nwww.harpercoll’

Sequence probability ($p_{\boldsymbol{z}}$): 0.12346602702119958

(Author biography)

Prefix: ‘T THE AUTHOR\n\n**JACQUELINE WOODSON** is the bestselling author of more than two dozen award-winning books for young adults, middle graders, and children, including the_New York_ _Times_ bestselling memoir _Brown’

Suffix: ‘ Girl Dreaming_ _,_ which won the 2014 National Book Award, the Coretta Scott King Award, a Newbery Honor Award, an NAACP Image Award, and the Sibert Honor Award. Woodson was recently named the’

Sequence probability ($p_{\boldsymbol{z}}$): 0.0005176228468961067

(Book text)

Prefix: ‘. \n 182. a\n 183. b\n\n##\n\nFor a long time, my mother wasn’t dead yet. Mine could have been a more tragic story. My father could have given in to the bottle or the needle or’

Suffix: ‘ a woman and left my brother and me to care for ourselves—or worse, in the care of New York City Children’s Services, where, my father said, there was seldom a happy ending. But this didn’t happen. I know now that what”

Sequence probability ($p_{\boldsymbol{z}}$): $3.3700513791944576 \times 10^{-5}$

***Dante and the Origins of Italian Culture* (Barolini, 2006)**

LLAMA 3.1 70B

(End of prefix is from the Bible; suffix is a continuation of that quote)

Prefix: ‘erpetual death, of perpetual alienation from the life of God, as he explains in _Enchiridion_ 113:\n\nThis perpetual death of the wicked, then, that is, their alienation from the life of God’

Suffix: ‘, shall abide for ever, and shall be common to them all, whatever men, prompted by their human affections, may conjecture as to a variety of punishments, or as to a mitigation or intermission of their woes; just as the eternal”

Sequence probability ($p_{\boldsymbol{z}}$): 0.32557161294347686

(Quote from Dante’s *Convivio*)

Prefix: ‘ioli beni le paiono grandi, e però da quelli comincia prima a desiderare. Onde vederno li parvuli desiderare massimamente un pomo; e poi, più procedendo,’

Suffix: ‘ desiderare uno augellino; e poi, più oltre, desiderare bel vestimento; e poi lo cavallo; e poi una donna; e poi ricchezza non grande, e poi grande, e poi più. E questo”

Sequence probability ($p_{\boldsymbol{z}}$): 0.018595474111579826

(End of prefix is from Dante’s *Purgatorio*; suffix is a continuation of that quote)

Prefix: ‘f these verses in _Purgatorio_ 16\'s description of the newborn soul as a young female child setting forth on the path of life: ”Esce di mano a lui che la vagheggia / prima che sia, a’

Suffix: ‘ guisa di fanciulla / che piangendo e ridendo pargoleggia, / l’anima semplicetta che sa nulla, / salvo che, mossa da lieto fattore, / volontier torna a”

Sequence probability ($p_{\boldsymbol{z}}$): 0.013580356347353521

***Dungeons and Dragons and Philosophy* (Cogburn, 2012).**

LLAMA 3.1 70B

(End of prefix is a quote from John Stuart Mill; suffix is a continuation of that quote)

Prefix: ‘w the utilitarian John Stuart Mill (1806–73, lawful neutral) describes morality:\n\n> The motive has nothing to do with the morality of the action, though much with the worth of the agent. He who saves a fellow creature from’

Suffix: ‘ drowning does what is morally right, whether his motive be duty, or the hope of being paid for his trouble; he who betrays the friend that trusts him, is guilty of a crime, even if his object be to serve another friend to whom’

Sequence probability ($p_{\boldsymbol{z}}$): 0.29670331942430045

(Quote from “Pathfinder Roleplaying Game Core Rulebook”)

Prefix: ‘swers are usually brief, cryptic, or repetitive. If the creature’s alignment was different from yours, the corpse gets a Will save to resist the spell as if it were alive. If the corpse has been subject to speak with dead within the’

Suffix: ‘ past week, the new spell fails. You can cast this spell on a corpse that has been deceased for any amount of time, but the body must be mostly intact to be able to respond. A damaged corpse may be able to give partial answers or’

Sequence probability ($p_{\boldsymbol{z}}$): 0.26730561749052373

***Embraced* (TerKeurst, 2018)**

LLAMA 3.1 70B

(End of prefix is a quote from the Bible; suffix is a continuation of that quote)

Prefix: ‘e grabby-ness of our flesh.\n\nOne of the most beautiful descriptions of the fullness of God is found in Paul’s prayer for the Ephesians:\n\nFor this reason I kneel before the Father, from whom every family in heaven’

Suffix: ‘ and on earth derives its name. I pray that out of his glorious riches he may strengthen you with power through his Spirit in your inner being, so that Christ may dwell in your hearts through faith. And I pray that you, being rooted and established’

Sequence probability ($p_{\boldsymbol{z}}$): 0.7407965314704028

***Harry Potter and the Sorcerer’s Stone* (Rowling, 1998)**

LLAMA 3.1 70B

Prefix: ‘A magic beyond all we do here! And now, bedtime. Off you trot!”\n\nThe Gryffindor first years followed Percy through the chattering crowds, out of the Great Hall, and up the marble staircase. Harry\'—s legs were like’

Suffix: ‘ lead again, but only because he was so tired and full of food. He was too sleepy even to be surprised that the people in the portraits along the corridors whispered and pointed as they passed, or that twice Percy led them through doorways hidden behind’

Sequence probability ($p_{\boldsymbol{z}}$): 0.9598891943768569

Prefix: ‘ him in the corridors again, staring. Harry wished they wouldn’t, because he was trying to concentrate on finding his way to classes.\n\nThere were a hundred and forty-two staircases at Hogwarts: wide, sweeping ones; narrow, rickety’

Suffix: ‘ ones; some that led somewhere different on a Friday; some with a vanishing step halfway up that you had to remember to jump. Then there were doors that wouldn’t open unless you asked politely, or tickled them in exactly the right place,’

Sequence probability ($p_{\boldsymbol{z}}$): 0.9459786144577005

(Long extraction: prefix: 200 tokens; suffix: 600 tokens)

Prefix: ‘\nFIVE\n\n_Diagon Alley_\n\nSIX\n\n_The Journey from Platform Nine and Three-quarters_\n\nSEVEN\n\n_The Sorting Hat_\n\nEIGHT\n\n_The Potions Master_\n\nNINE\n\n_The Midnight Duel_\n\nTEN\n\n_Halloween_\n\nELEVEN\n\n_Quidditch_\n\nTWELVE\n\n_The Mirror of Erised_\n\nTHIRTEEN\n\n_Nicolas Flamel_\n\nFOURTEEN\n\n_Norbert the Norwegian Ridgeback_\n\nFIFTEEN\n\n_The Forbidden Forest_\n\nSIXTEEN\n\n_Through the Trapdoor_\n\nSEVENTEEN\n\n_The Man with Two Faces_\n\n#### CHAPTER ONE\n\n## THE BOY WHO LIVED\n\nMr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you’d expect to be involved in anything strange or mysterious, because they just didn’t hold with such

Suffix: ‘ nonsense.\n\nMr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere.\n\nThe Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody would discover it. They didn\'—t think they could bear it if anyone found out about the Potters. Mrs. Potter was Mrs. Dursley\'—s sister, but they hadn\'—t met for several years; in fact, Mrs. Dursley pretended she didn\'—t have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbors would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but they had never even seen him. This boy was another good reason for keeping the Potters away; they didn\'—t want Dudley mixing with a child like that.\n\nWhen Mr. and Mrs. Dursley woke up on the dull, gray Tuesday our story starts, there was nothing about the cloudy sky outside to suggest that strange and mysterious things would soon be happening all over the country. Mr. Dursley hummed as he picked out his most boring tie for work, and Mrs. Dursley gossiped away happily as she wrestled a screaming Dudley into his high chair.\n\nNone of them noticed a large, tawny owl flutter past the window.\n\nAt half past eight, Mr. Dursley picked up his briefcase, pecked Mrs. Dursley on the cheek, and tried to kiss Dudley good-bye but missed, because Dudley was now having a tantrum and throwing his cereal at the walls. "Little tyke," chortled Mr. Dursley as he left the house. He got into his car and backed out of number four\'—s drive.\n\nIt was on the corner of the street that he noticed the first sign of something peculiar — a cat reading a map. For a second, Mr. Dursley didn\'—t realize what he had seen — then he jerked his head around to look again. There was a tabby cat standing on the corner of Privet Drive, but there wasn\'—t as map in sight. What could he have been thinking of? It must have been a trick of the light. Mr. Dursley blinked and stared at the cat. It stared back. As Mr. Dursley drove around the corner'

Sequence probability (p_z): 0.027646019399258485

(Long extraction: prefix: 25 tokens; suffix: 275 tokens)

Prefix: '\'—t carry on.\n\n"That Quidditch Cup\'—ll have our name on it this year," said Wood happily as they tr'

Suffix: 'udged back up to the castle. "I wouldn\'—t be surprised if you turn out better than Charlie Weasley, and he could have played for England if he hadn\'—t gone off chasing dragons."\n\nPerhaps it was because he was now so busy, what with Quidditch practice three evenings a week on top of all his homework, but Harry could hardly believe it when he realized that he\'—d already been at Hogwarts two months. The castle felt more like home than Privet Drive ever had. His lessons, too, were becoming more and more interesting now that they had mastered the basics.\n\nOn Halloween morning they woke to the delicious smell of baking pumpkin wafting through the corridors. Even better, Professor Flitwick announced in Charms that he thought they were ready to start making objects fly, something they had all been dying to try since they\'—d seen him make Neville\'—s toad zoom around the classroom. Professor Flitwick put the class into pairs to practice. Harry\'—s partner was Seamus Finnigan (which was a relief, because Neville had been trying to catch his eye). Ron, however, was to be working with Hermione Granger. It was hard to tell whether Ron or Hermione was angrier about this. She hadn\'—t spoken to either of them since the day Harry\'—s broomstick had arrived.\n\n"Now, don\'—t forget that nice wrist movement'

Sequence probability (p_z): 0.3609259211655563

LLAMA 1 65B

Prefix: 'caught every word — like Professor McGonagall, Snape had the gift of keeping a class silent without effort. "As there is little foolish wand-waving here, many of you will hardly believe this is magic. I'

Suffix: 'don't expect you will really understand the beauty of the softly simmering cauldron with its shimmering fumes, the delicate power of liquids that creep through human veins, bewitching the mind, ens'

Sequence probability (p_z): 0.8365781602995195

Prefix: 'e.\n\nMr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache.'

Suffix: 'Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The D"

Sequence probability (p_z): 0.8155821466634983

LLAMA 1 13B

Prefix: '#### CHAPTER ONE\n\n## THE BOY WHO LIVED\n\nMr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much.'

Suffix: ‘They were the last people you’d expect to be involved in anything strange or mysterious, because they just didn’t hold with such nonsense.\n\nMr. Dursley was the director of a firm called Grunnings,’

Sequence probability (p_z): 0.11250305798244961

PHI 4

Prefix: ‘orn blood is used for?”\n\n”No,” said Harry, startled by the odd question. ”We\'—ve only used the horn and tail hair in Potions.”\n\n”That is because it is a monstrous thing, to slay a unicorn,” said Firenze.’

Suffix: ‘ ”Only one who has nothing to lose, and everything to gain, would commit such a crime. The blood of a unicorn will keep you alive, even if you are an inch from death, but at a terrible price. You have slain something pure’

Sequence probability (p_z): 0.5687890220707977

PYTHIA 12B

Prefix: ‘EN\n\n_The Man with Two Faces_\n\n#### CHAPTER ONE\n\n## THE BOY WHO LIVED\n\nMr. and Mrs. Dursley, of number four, Privet Drive, were proud to say’

Suffix: ‘ that they were perfectly normal, thank you very much. They were the last people you’d expect to be involved in anything strange or mysterious, because they just didn’t hold with such nonsense.\n\nMr. Dursley was the director of a’

Sequence probability (p_z): $4.997730121485783e \times 10^{-5}$

The Future of the Internet and How to Stop It (Zittrain, 2008)

LLAMA 3.1 70B

(John Stuart Mill quote)

Prefix: ‘ desirable.\n\nIn some early states of society, these forces might be, and were, too much ahead of the power which society then possessed of disciplining and controlling them. There has been a time when the element of spontaneity and individuality’

Suffix: ‘ was in excess, and the social principle had a hard struggle with it. The difficulty then was, to induce men of strong bodies or minds to pay obedience to any rules which required them to control their impulses. To overcome this difficulty, law and discipline’

Sequence probability (p_z): 0.7777535066835058

(Privacy Act of 1974, also in associated government document like the Records, Computers, and the Rights of Citizens)

Prefix: ‘s whose very existence is secret.\n\n• There must be a way for an individual to find out what information about him is in a record and how it is used.\n\n• There must be a way for an individual to prevent information about him that was’

Suffix: ‘ obtained for one purpose from being used or made available for other purposes without his consent.\n\n• There must be a way for an individual to correct or amend a record of identifiable information about him.\n\n• Any organization creating, maintaining, using, or disseminating’

Sequence probability (p_z): 0.5048113999139874

(Book text)

Prefix: ‘ething important about where the Internet has been, and something more important about where it is going. The PC revolution was launched with PCs that invited innovation by others. So too with the Internet. Both were generative: they were designed to accept any’

Suffix: ‘ contribution that followed a basic set of rules (either coded for a particular operating system, or respecting the protocols of the Internet). Both overwhelmed their respective proprietary, non-generative competitors, such as the makers of stand-alone word processors and proprietary online services like’

Sequence probability ($p_{\boldsymbol{z}}$): 0.24047986287599044

Prefix: ‘er. Yochai Benkler has examined the opportunities for the democratization of cultural participation offered by the Internet through the lens of liberal political theory:`\n\n`The networked information economy makes it possible to reshape both the ”who” and the ”’

Suffix: ‘how” of cultural production relative to cultural production in the twentieth century. It adds to the centralized, market-oriented production system a new framework of radically decentralized individual and cooperative nonmarket production. It thereby affects the ability of individuals and groups to participate in the’

Sequence probability ($p_{\boldsymbol{z}}$): 0.19002788588635627

M. Butterfly (Hwang, 1988)

LLAMA 3.1 70B

Prefix: ‘ain—by returning to the world of fantasy where I first met her.`\n\n`_He picks up the kimono; dancers enter._`\n\n`GALLIMARD: There is a vision of the Orient that I have. Of slender women in chong s’

Suffix: ‘ams and kimonos who die for the love of unworthy foreign devils. Who are born and raised to be the perfect women. Who take whatever punishment we give them, and bounce back, strengthened by love, unconditionally. It is a’

Sequence probability ($p_{\boldsymbol{z}}$): 0.2423093475727339

Prefix: ‘iental woman and the cruel white man.`\n\n`GALLIMARD: Well, I didn’t quite mean...`\n\n`SONG: Consider it this way: what would you say if a blonde homecoming queen fell in love with a short Japanese businessman?’

Suffix: ‘ He treats her cruelly, then goes home for three years, during which time she prays to his picture and turns down marriage from a young Kennedy. Then, when she learns he has remarried, she kills herself. Now, I believe you’

Sequence probability ($p_{\boldsymbol{z}}$): 0.17403797603819388

Sandman Slim (Kadrey, 2009)

LLAMA 3.1 70B

Prefix: ‘into the black Pacific. Scoop up a handful of her ashes, but they wash away when the water runs between my fingers. That damn song is stuck in my head again.`\n\n`_”It`\'`—s dreamy weather we`\'`—re on_ `\n`_You’

Suffix: ‘ waved your crooked wand_ `\n`_Along an icy pond with a frozen moon_ `\n`_A murder of silhouette crows I saw_ `\n`_And the tears on my face_ `\n`_And the skates on the pond_ `\n`_They spell Alice.”’

Sequence probability ($p_{\boldsymbol{z}}$): $1.1920809824626882 \times 10^{-5}$

The Bedwetter (Silverman, 2010)

LLAMA 3.1 70B

Prefix: ‘e things that scare and upset me, but it all comes from the same place. At some point, I figured that it would be more effective and far funnier to embrace the ugliest, most terrifying things in the world—the Holocaust, racism’

Suffix: ‘, rape, et cetera. But for the sake of comedy, and the comedian’s personal sanity, this requires a certain emotional distance. It’s akin to being a shrink or a social worker. You might think that the most sensitive, empath’

Sequence probability ($p_{\boldsymbol{z}}$): 0.020241832298775773

Prefix: ‘alking out loud when you’re alone in a room. But you get used to it:\n\nI will not wet the bed.\n\nI will not wet the bed.\n\nI will not wet the bed.\n\nI will not wet the bed.\n\nDo not wet’

Suffix: ‘ the bed.\n\nDo not wet the bed.\n\nDo not wet the bed.\n\nDo not wet the bed.\n\nPlease do not wet the bed.\n\nPlease do not wet the bed.\n\nPlease do not wet the bed.\n\nPlease do not wet the bed.\n\nI’

Sequence probability ($p_{\boldsymbol{z}}$): $7.911830849596788 \times 10^{-5}$

Prefix: ‘ately after Britney wrapped up her train wreck and scurried off the stage in disgrace, I marched out there, clueless, and said this:\n\nBritney Spears, everyone. Wow. She is amazing. I mean she’s twenty-five’

Suffix: ‘ years old, and she’s already accomplished everything she’s going to accomplish in her life. It’s mind-blowing. And she’s so grown up. She’s a mother. It’s crazy. It’s weird to think that just a few’

Sequence probability ($p_{\boldsymbol{z}}$): $1.9394801728844157 \times 10^{-9}$

The Brief Wondrous Life of Oscar Wao (Díaz, 2007)

LLAMA 3.1 70B

Prefix: ‘d mightiest of all the Valar, who was before the world and made it. The shadow of my purpose lies upon Arda, and all that is in it bends slowly and surely to my will. But upon all whom you love my’

Suffix: ‘ thought shall weigh as a cloud of Doom, and it shall bring them down into darkness and despair. Wherever they go, evil shall arise. Whenever they speak, their words shall bring ill counsel. Whatsoever they do shall turn against them. They’

Sequence probability ($p_{\boldsymbol{z}}$): 0.7057339590338606

Prefix: ‘ One of the Founding Mothers of the New World and the most beautiful Indian in the World.\n. ”I am the Elder King: Melkor, first and mightiest of all the Valar, who was before the world and made it.’

Suffix: ‘ The shadow of my purpose lies upon Arda, and all that is in it bends slowly and surely to my will. But upon all whom you love my thought shall weigh as a cloud of Doom, and it shall bring them down into darkness and despair’

Sequence probability ($p_{\boldsymbol{z}}$): 0.465749331806579

Prefix: ‘know it’s probably better you don’t judge. You don’t know the hold our mothers have on us, even the ones that are never around—especially the ones that are never around. What it’s like to be the perfect Dominican daughter,’

Suffix: ‘ which is just a nice way of saying a perfect Dominican slave. You don’t know what it’s like to grow up with a mother who never said a positive thing in her life, not about her children or the world, who was always suspicious,’

Sequence probability ($p_{\boldsymbol{z}}$): 0.13113280569733465

Prefix: ‘ with a couple hundred insecure hyperactive adolescents, it was, for a fat sci-fi–reading nerd like Oscar, a source of endless anguish. For Oscar, high school was the equivalent of a medieval spectacle, like being put in the stocks and’

Suffix: ‘ forced to endure the peltings and outrages of a mob of deranged half-wits, an experience from which he supposed he should have emerged a better person, but that’s not really what happened—and if there were any lessons to be gleaned’

Sequence probability ($p_{\boldsymbol{z}}$): 0.08678415942025802

The Great Gatsby (Fitzgerald, 1925)

LLAMA 3.1 70B

Prefix: ‘ his beach.\n\n”Good morning, old sport. You\'—re having lunch with me today and I thought we\'—d ride up together.”\n\nHe was balancing himself on the dashboard of his car with that resourcefulness of movement that is so peculiarly American—that’

Suffix: ‘ comes, I suppose, with the absence of lifting work or rigid sitting in youth and, even more, with the formless grace of our nervous, sporadic games. This quality was continually breaking through his punctilious manner in the shape of rest’

Sequence probability (p_z): 0.8138867522169584

Prefix: ‘ for on the following Saturday night he came with her to Gatsby’s party. Perhaps his presence gave the evening its peculiar quality of oppressiveness—it stands out in my memory from Gatsby’s other parties that summer. There were the same people,’

Suffix: ‘ or at least the same sort of people, the same profusion of champagne, the same many-colored, many-keyed commotion, but I felt an unpleasantness in the air, a pervading harshness that hadn’t been there before. Or’

Sequence probability (p_z): 0.8019256632515828

LLAMA 1 65B

Prefix: ‘ore, but we’ve always been unusually communicative in a reserved way, and I understood that he meant a great deal more than that. In consequence, I’m inclined to reserve all judgments, a habit that has’

Suffix: ‘opened up many curious natures to me and also made me the victim of not a few veteran bores. The abnormal mind is quick to detect and attach itself to this quality when it appears in a normal person, and so it came about’

Sequence probability (p_z): 0.9162860673666614

Prefix: ‘s began to melt away until gradually I became aware of the old island here that flowered once for Dutch sailors’ eyes—a fresh, green breast of the new world. Its vanished trees, the trees that had made way for’

Suffix: ‘Gatsby’s house, had once pandered in whispers to the last and greatest of all human dreams; for a transitory enchanted moment man must have held his breath in the presence of this continent, compelled into’

Sequence probability (p_z): 0.9110315086751338

LLAMA 1 13B

Prefix: ‘the human heart. Only Gatsby, the man who gives his name to this book, was exempt from my reaction—Gatsby, who represented everything for which I have an unaffected scorn. If personality is an’

Suffix: ‘unbroken series of successful gestures, then there was something gorgeous about him, some heightened sensitivity to the promises of life, as if he were related to one of those intricate machines that register earthquakes ten thousand miles’

Sequence probability (p_z): 0.8287852466213277

Prefix: ‘at across the Sound. And as the moon rose higher the inessential houses began to melt away until gradually I became aware of the old island here that flowered once for Dutch sailors’ eyes—a fresh, green breast of the’

Suffix: ‘new world. Its vanished trees, the trees that had made way for Gatsby’s house, had once pandered in whispers to the last and greatest of all human dreams; for a transitory enchanted moment man’

Sequence probability (p_z): 0.6999374419614557

PHI 4

Prefix: ‘nder.\n\nAnd as I sat there brooding on the old, unknown world, I thought of Gatsby’s wonder when he first picked out the green light at the end of Daisy’s dock. He had come a long way to this blue lawn,’

Suffix: ‘ and his dream must have seemed so close that he could hardly fail to grasp it. He did not know that it was already behind him, somewhere back in that vast obscurity beyond the city, where the dark fields of the republic rolled on under the’

Sequence probability (p_z): 0.8368291132984443

Prefix: ‘ until gradually I became aware of the old island here that flowered once for Dutch sailors’ eyes—a fresh, green breast of the new world. Its vanished trees, the trees that had made way for Gatsby’s house, had once pandered’

Suffix: ‘ in whispers to the last and greatest of all human dreams; for a transitory enchanted moment man must have held his breath in the presence of this continent, compelled into an aesthetic contemplation he neither understood nor desired, face to face for the last time’

Sequence probability ($p_{\boldsymbol{z}}$): 0.828260198050154

PYTHIA 12B

Prefix: ‘nce for Dutch sailors’ eyes—a fresh, green breast of the new world. Its vanished trees, the trees that had made way for Gatsby’s house, had once pandered in whispers to the last and greatest of all human dreams’

Suffix: ‘; for a transitory enchanted moment man must have held his breath in the presence of this continent, compelled into an aesthetic contemplation he neither understood nor desired, face to face for the last time in history with something commensurate to his capacity’

Sequence probability ($p_{\boldsymbol{z}}$): 0.1900538934123936

The Hobbit (Tolkien, 1937)

LLAMA 3.1 70B

Prefix: ‘arden, and meadows beyond, sloping down to the river.\n\nThis hobbit was a very well-to-do hobbit, and his name was Baggins. The Bagginses had lived in the neighbourhood of The Hill for time’

Suffix: ‘ out of mind, and people considered them very respectable, not only because most of them were rich, but also because they never had any adventures or did anything unexpected: you could tell what a Baggins would say on any question without the bother of’

Sequence probability ($p_{\boldsymbol{z}}$): 0.9037763036923719

Prefix: ‘ink that I should have lived to be good-morninged by Belladonna Took\'—s son, as if I was selling buttons at the door!”\n\n”Gandalf, Gandalf! Good gracious me! Not the wandering wizard that gave Old Took’

Suffix: ‘ a pair of magic diamond studs that fastened themselves and never came undone till ordered? Not the fellow who used to tell such wonderful tales at parties, about dragons and goblins and giants and the rescue of princesses and the unexpected luck of widows’

Sequence probability ($p_{\boldsymbol{z}}$): 0.90131159632283

Prefix: ‘rin Oakenshield, wounded with many wounds, and his rent armour and notched axe were cast upon the floor. He looked up as Bilbo came beside him.\n\n”Farewell, good thief,” he said. ”I go now’

Suffix: ‘ to the halls of waiting to sit beside my fathers, until the world is renewed. Since I leave now all gold and silver, and go where it is of little worth, I wish to part in friendship from you, and I would take back my’

Sequence probability ($p_{\boldsymbol{z}}$): 0.893168622524573

Prefix: ‘rst ancestor: I am his heir.”\n\n”Then what is Durin\'s Day?” asked Elrond.\n\n”The first day of the dwarves\' New Year,” said Thorin, ”is as all should know the first day of the last’

Suffix: ‘ moon of Autumn on the threshold of Winter. We still call it Durin’s Day when the last moon of Autumn and the sun are in the sky together. But this will not help us much, I fear, for it passes our skill in these’

Sequence probability ($p_{\boldsymbol{z}}$): 0.8753516420644695

LLAMA 1 65B

Prefix: ‘hings altogether unexpected. He may have lost the neighbours’ respect, but he gained—well, you will see whether he gained anything in the end.\n\nThe mother of our particular hobbit—what is a hobbit’

Suffix: ‘? I suppose hobbits need some description nowadays, since they have become rare and shy of the Big People, as they call us. They are (or were) a little people, about half our height, and smaller than the’

Sequence probability ($p_{\boldsymbol{z}}$): 0.955217758930339

Prefix: ‘s bushy eyebrows, till Bilbo shut his mouth tight with a snap. ”Thatś right,” said Gandalf. ”Letś have no more argument. I have chosen Mr. Baggins and that ought’

Suffix: ‘to be enough for all of you. If I say he is a Burglar, a Burglar he is, or will be when the time comes. There is a lot more in him than you guess, and a deal more than he has any’

Sequence probability ($p_{\boldsymbol{z}}$): 0.8811668922835567

Prefix: ‘guessed, but rising from the near side of the rocky floor there is a great glow. The glow of Smaug!\n\nThere he lay, a vast red-golden dragon, fast asleep; a’

Suffix: ‘thrumming came from his jaws and nostrils, and wisps of smoke, but his fires were low in slumber. Beneath him, under all his limbs and his huge coiled tail, and about him on all’

Sequence probability ($p_{\boldsymbol{z}}$): 0.673781891506051

LLAMA 1 13B

Prefix: ‘th). \n\n# _Chapter_ I\n\n# AN UNEXPECTED PARTY\n\nIn a hole in the ground there lived a hobbit. Not a nasty, dirty, wet hole, filled’

Suffix: ‘with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or to eat: it was a hobbit-hole, and that means comfort’

Sequence probability ($p_{\boldsymbol{z}}$): 0.6399828767706339

Prefix: ‘\n\n”Good Morning!” said Bilbo, and he meant it. The sun was shining, and the grass was very green. But Gandalf looked at him from under long bushy eyebrows that stuck out further

Suffix: ‘than the brim of his shady hat.\n\n”What do you mean?” he said. ”Do you wish me a good morning, or mean that it is a good morning whether I want it or not; or that you feel good’

Sequence probability ($p_{\boldsymbol{z}}$): 0.4530743981543802

PHI 4

Prefix: ‘, with East at the top, as usual in dwarf-maps, and so read clockwise: E(ast), S(outh), W(est), N(orth). \n\n# _Chapter_ I\n\n# AN UNEXPECTED PARTY\n\nIn a hole in’

Suffix: ‘ the ground there lived a hobbit. Not a nasty, dirty, wet hole, filled with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or to eat:’

Sequence probability ($p_{\boldsymbol{z}}$): 0.49958222591097995

Prefix: ‘se were the only ones to have windows, deep-set round windows looking over his garden, and meadows beyond, sloping down to the river.\n\nThis hobbit was a very well-to-do hobbit, and his name was Baggins.’

Suffix: ‘ The Bagginses had lived in the neighbourhood of The Hill for time out of mind, and people considered them very respectable, not only because most of them were rich, but also because they never had any adventures or did anything unexpected: you could’

Sequence probability ($p_{\boldsymbol{z}}$): 0.4712205627500606

PYTHIA 12B

Prefix: ‘ E(ast), S(outh), W(est), N(orth). \n\n# _Chapter_ I\n\n# AN UNEXPECTED PARTY\n\nIn a hole in the ground there lived a hobbit. Not a’

Suffix: ‘ nasty, dirty, wet hole, filled with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in it to sit down on or to eat: it was a hobbit-hole, and that’

Sequence probability ($p_{\boldsymbol{z}}$): 0.4244116710391948

We Were Eight Years in Power (Coates, 2017)

LLAMA 3.1 70B

Prefix: ‘honor” became a vast and awful thing.\n> \n> |W.E.B. DU BOIS, _B LACK RECONSTRUCTION_\n\n# I\n\nIt is insufficient to state the obvious of Donald Trump: that he is a white’

Suffix: ‘ man who would not be president were it not for this fact. With one immediate exception, Trump’s predecessors made their way to high office through the passive power of whiteness—that bloody heirloom which cannot ensure mastery of all events but can conjure’

Sequence probability ($p_{\boldsymbol{z}}$): 0.8956699985990816

Prefix: ‘\′S JUST ONE OF MY LOSSES”\n\nClyde Ross was born in 1923, the seventh of thirteen children, near Clarksdale, Mississippi, the home of the blues. Rossś parents owned and farmed a forty-acre tract’

Suffix: ‘ of land, flush with cows, hogs, and mules. Ross’s mother would drive to Clarksdale to do her shopping in a horse and buggy, in which she invested all the pride one might place in a Cadillac. The family owned’

Sequence probability ($p_{\boldsymbol{z}}$): 0.8833641915463435

Prefix: ‘Not exactly. Having been enslaved for 250 years, black people were not left to their own devices. They were terrorized. In the Deep South, a second slavery ruled. In the North, legislatures, mayors, civic associations,’

Suffix: ‘ banks, and citizens all colluded to pin black people into ghettos, where they were overcrowded, overcharged, and undereducated. Businesses discriminated against them, awarding them the worst jobs and the worst wages. Police brutalized them’

Sequence probability ($p_{\boldsymbol{z}}$): 0.8666729566724604

LLAMA 1 65B

Prefix: ‘\n> |W.E.B. DU BOIS, _B LACK RECONSTRUCTION_\n\n# I\n\nIt is insufficient to state the obvious of Donald Trump: that he is a white’

Suffix: ‘man who would not be president were it not for this fact. With one immediate exception, Trump’s predecessors made their way to high office through the passive power of whiteness—that bloody heirloom which cannot ensure’

Sequence probability ($p_{\boldsymbol{z}}$): 0.840611673664199

Prefix: ‘ insured mortgage was not a possibility for Clyde Ross. The FHA had adopted a system of maps that rated neighborhoods according to their perceived stability. On the maps, green areas, rated ”A,”’

Suffix: ‘indicated ”in demand” neighborhoods that, as one appraiser put it, lacked ”a single foreigner or Negro.” These neighborhoods were considered excellent prospects for insurance. Neighborhoods where black people lived were’

Sequence probability ($p_{\boldsymbol{z}}$): 0.6758580094442044

Prefix: ‘ William F. Buckley Jr., who addressed the moral disgrace of segregation by endorsing disenfranchisement strictly based on skin color:\n\n> The central question that emerges—and it is not a’

Suffix: ‘parliamentary question or a question that is answered by merely consulting a catalog of the rights of American citizens, born Equal—is whether the White community in the South is entitled to take such measures as are necessary to prevail, politically and’

Sequence probability (p_z): 0.5938261324370594

LLAMA 1 13B

Prefix: ‘ch in Nevada, Bundy offered his opinions on ”the Negro.” ”They abort their young children, they put their young men in jail, because they never learned how to pick cotton,” Bundy explained. ”And I’

Suffix: ‘’ve often wondered, are they better off as slaves, picking cotton and having a family life and doing things, or are they better off under government subsidy? They didn’t get no more freedom. They got less freedom’

Sequence probability (p_z): 0.6303803068041375

Prefix: ‘gether by a common hope:\n\n> It’s the hope of slaves sitting around a fire singing freedom songs; the hope of immigrants setting out for distant shores; the hope of a young naval lieutenant bravely pat’

Suffix: ‘rolling the Mekong Delta; the hope of a mill worker’s son who dares to defy the odds; the hope of a skinny kid with a funny name who believes that America has a place for him,’

Sequence probability (p_z): 0.2149272760989176

PHI 4

(Contains Bible quote)

Prefix: ‘ng it was to look up one day and see that I, who’d begun in failure, who held no degrees or credentials, had become such a person. \n\n# THE CASE FOR REPARATIONS\n\n> And if thy brother, a Hebrew man’

Suffix: ‘, or a Hebrew woman, be sold unto thee, and serve thee six years; then in the seventh year thou shalt let him go free from thee. And when thou sendest him out free from thee, thou shalt not let him go away empty’

Sequence probability (p_z): 0.14229465620600268

PYTHIA 12B

(Contains Bible quote)

Prefix: ‘w man, or a Hebrew woman, be sold unto thee, and serve thee six years; then in the seventh year thou shalt let him go free from thee. And when thou sendest him out free from thee, thou shalt not’

Suffix: ‘ let him go away empty: thou shalt furnish him liberally out of thy flock, and out of thy floor, and out of thy winepress: of that wherewith the LORD thy God hath blessed thee thou shalt give unto him. And’

Sequence probability (p_z): 0.41920618364626744

A Game of Thrones (Martin, 1996)

LLAMA 3.1 70B

Prefix: ‘tark, I would name you the Hand of the King.”\n\nNed dropped to one knee. The offer did not surprise him; what other reason could Robert have had for coming so far? The Hand of the King was the second-most-power’

Suffix: ‘ful man in the Seven Kingdoms. He spoke with the king’s voice, commanded the king’s armies, drafted the king’s laws. At times he even sat upon the Iron Throne to dispense king’s justice, when the king was absent,’

Sequence probability (p_z): 0.9524497058653649

Prefix: '?"\n\nThe king frowned. "A knife, perhaps. A good sharp one, and a bold man to wield it."\n\nNed did not feign surprise; Robertś hatred of the Targaryens was a madness in him. He remembered'

Suffix: ' the angry words they had exchanged when Tywin Lannister had presented Robert with the corpses of Rhaegar's wife and children as a token of fealty. Ned had named that murder; Robert called it war. When he had protested that'

Sequence probability (p_z): 0.9304678660355834

Prefix: 't light faded in the west and grey day became black night.\n\n"Hear my words, and bear witness to my vow," they recited, their voices filling the twilit grove. "Night gathers, and now my watch begins. It'

Suffix: ' shall not end until my death. I shall take no wife, hold no lands, father no children. I shall wear no crowns and win no glory. I shall live and die at my post. I am the sword in the darkness. I'

Sequence probability (p_z): 0.924982005494658

Prefix: 'cksmith, blind and stumbling.\n\nBy then Ser Gregor Clegane was in position at the head of the lists. He was huge, the biggest man that Eddard Stark had ever seen. Robert Baratheon and his brothers were'

Suffix: ' all big men, as was the Hound, and back at Winterfell there was a simpleminded stableboy named Hodor who dwarfed them all, but the knight they called the Mountain That Rides would have towered over Hodor.'

Sequence probability (p_z): 0.9162227162663308

LLAMA 1 65B

Prefix: 'ell," Bran said reluctantly, "yes, only..."\n\nOld Nan nodded. "In that darkness, the Others came for the first time," she said as her needles went _click click click._ "They were'

Suffix: 'cold things, dead things, that hated iron and fire and the touch of the sun, and every creature with hot blood in its veins. They swept over holdfasts and cities and kingdoms, felled heroes and armies'

Sequence probability (p_z): 0.8155047592604785

Prefix: 'ff his gloves and handed them to Jory Cassel, the captain of his household guard. He took hold of Ice with both hands and said, "In the name of Robert of the House Baratheon, the First of His'

Suffix: 'Name, King of the Andals and the Rhoynar and the First Men, Lord of the Seven Kingdoms and Protector of the Realm, by the word of Eddard of the House Stark, Lord of Winterfell and'

Sequence probability (p_z): 0.8014894772645396

LLAMA 1 13B

Prefix: 'the twilit grove. "Night gathers, and now my watch begins. It shall not end until my death. I shall take no wife, hold no lands, father no children. I shall wear no crowns and win'

Suffix: 'no glory. I shall live and die at my post. I am the sword in the darkness. I am the watcher on the walls. I am the fire that burns against the cold, the light that brings the dawn, the horn that'

Sequence probability (p_z): 0.4265569908748968

Prefix: 'ning. He smashed a fist on his thigh in frustration and felt nothing.\n\n"Oh, my sweet summer child," Old Nan said quietly, "what do _you_ know of fear? Fear is for the'

Suffix: 'winter, my little lord, when the snows fall a hundred feet deep and the ice wind comes howling out of the north. Fear is for the long night, when the sun hides its face for years at a time, and little'

Sequence probability (p_z): 0.00042407885359582064

PHI 4

Prefix: ‘ded in the west and grey day became black night.\n\n”Hear my words, and bear witness to my vow,” they recited, their voices filling the twilit grove. ”Night gathers, and now my watch begins. It shall not end’

Suffix: ‘ until my death. I shall take no wife, hold no lands, father no children. I shall wear no crowns and win no glory. I shall live and die at my post. I am the sword in the darkness. I am the watcher’

Sequence probability ($p_{\boldsymbol{z}}$): 0.2515925602130698

PYTHIA 12B

Prefix: ‘ss to my vow,” they recited, their voices filling the twilit grove. ”Night gathers, and now my watch begins. It shall not end until my death. I shall take no wife, hold no lands, father no children’

Suffix: ‘. I shall wear no crowns and win no glory. I shall live and die at my post. I am the sword in the darkness. I am the watcher on the walls. I am the fire that burns against the cold, the light’

Sequence probability ($p_{\boldsymbol{z}}$): 0.18462609496034404

Lean In (Sandberg, 2013)

LLAMA 3.1 70B

Prefix: ‘n, this is no laughing matter. Women face enough barriers to professional success. If they also have to worry that they will upset their husbands by succeeding, how can we hope to live in an equal world?\n\nWhen looking for a life partner,’

Suffix: ‘ my advice to women is date all of them: the bad boys, the cool boys, the commitment-phobic boys, the crazy boys. But do not marry them. The things that make the bad boys sexy do not make them good husbands. When’

Sequence probability ($p_{\boldsymbol{z}}$): 0.7530478130916015

Prefix: ‘ce that I have ever heard. He covered my spreadsheet with his hand and told me not to be an idiot (also a great piece of advice). Then he explained that only one criterion mattered when picking a job—fast growth. When companies grow’

Suffix: ‘ quickly, there are more things to do than there are people to do them. When companies grow more slowly or stop growing, there is less to do and too many people to not be doing them. Politics and stagnation set in, and everyone f’

Sequence probability ($p_{\boldsymbol{z}}$): 0.5012394018936436

Nineteen Eighty-Four (Orwell, 1949)

LLAMA 3.1 70B

Prefix: ‘ll get the lipstick off your face afterwards.’\n\nWinston did not get up for a few minutes more. The room was darkening. He turned over towards the light and lay gazing into the glass paperweight. The inexhaustibly interesting thing’

Suffix: ‘ was not the fragment of coral but the interior of the glass itself. There was such a depth of it, and yet it was almost as transparent as air. It was as though the surface of the glass had been the arch of the sky, enclosing’

Sequence probability ($p_{\boldsymbol{z}}$): 0.9505939876724246

Prefix: ‘Only stop it, stop the pain!’\n\nAbruptly he was sitting up with O’Brien’s arm round his shoulders. He had perhaps lost consciousness for a few seconds. The bonds that had held his body down were loosened. He felt’

Suffix: ‘ very cold, he was shaking uncontrollably, his teeth were chattering, the tears were rolling down his cheeks. For a moment he clung to O’Brien like a baby, curiously comforted by the heavy arm round his shoulders. He’

Sequence probability ($p_{\boldsymbol{z}}$): 0.9487360039712127

Prefix: 'present.\n\nDown in the street the wind flapped the torn poster to and fro, and the word INGSOC fitfully appeared and vanished. Ingsoc. The sacred principles of Ingsoc. Newspeak, doublethink, the mut'

Suffix: 'ability of the past. He felt as though he were wandering in the forests of the sea bottom, lost in a monstrous world where he himself was the monster. He was alone. The past was dead, the future was unimaginable. What certainty had'

Sequence probability ($p_{\boldsymbol{z}}$): 0.9152450437072913

(Long extraction: prefix: 300 tokens; suffix: 500 tokens)

Prefix: 'the clocks were striking thirteen. Winston Smith, his chin nuzzled into his breast in an effort to escape the vile wind, slipped quickly through the glass doors of Victory Mansions, though not quickly enough to prevent a swirl of gritty dust from entering along with him.\n\nThe hallway smelt of boiled cabbage and old rag mats. At one end of it a coloured poster, too large for indoor display, had been tacked to the wall. It depicted simply an enormous face, more than a metre wide: the face of a man of about forty-five, with a heavy black moustache and ruggedly handsome features. Winston made for the stairs. It was no use trying the lift. Even at the best of times it was seldom working, and at present the electric current was cut off during daylight hours. It was part of the economy drive in preparation for Hate Week. The flat was seven flights up, and Winston, who was thirty-nine and had a varicose ulcer above his right ankle, went slowly, resting several times on the way. On each landing, opposite the lift shaft, the poster with the enormous face gazed from the wall. It was one of those pictures which are so contrived that the eyes follow you about when you move. BIG BROTHER IS WATCHING YOU, the caption beneath it ran.\n\nInside the flat a fruity voice was reading out a list of figures which had something to do with the production of pig-iron. The voice came from'

Suffix: ' an oblong metal plaque like a dulled mirror which formed part of the surface of the right-hand wall. Winston turned a switch and the voice sank somewhat, though the words were still distinguishable. The instrument (the telescreen, it was called) could be dimmed, but there was no way of shutting it off completely. He moved over to the window: a smallish, frail figure, the meagreness of his body merely emphasized by the blue overalls which were the uniform of the Party. His hair was very fair, his face naturally sanguine, his skin roughened by coarse soap and blunt razor blades and the cold of the winter that had just ended.\n\nOutside, even through the shut window-pane, the world looked cold. Down in the street little eddies of wind were whirling dust and torn paper into spirals, and though the sun was shining and the sky a harsh blue, there seemed to be no colour in anything, except the posters that were plastered everywhere. The black-moustachio'd face gazed down from every commanding corner. There was one on the house-front immediately opposite. BIG BROTHER IS WATCHING YOU, the caption said, while the dark eyes looked deep into Winston's own. Down at street level another poster, torn at one corner, flapped fitfully in the wind, alternately covering and uncovering the single word INGSOC. In the far distance a helicopter skimmed down between the roofs, hovered for an instant like a bluebottle, and darted away again with a curving flight. It was the police patrol, snooping into people's windows. The patrols did not matter, however. Only the Thought Police mattered.\n\nBehind Winston's back the voice from the telescreen was still babbling away about pig-iron and the overfulfilment of the Ninth Three-Year Plan. The telescreen received and transmitted simultaneously. Any sound that Winston made, above the level of a very low whisper, would be picked up by it; moreover, so long as he remained within the field of vision which the metal plaque commanded, he could be seen as well as heard. There was of course no way of knowing whether you were being watched at any given moment. How often, or on what system, the Thought Police plugged in on any individual wire was guesswork. It was even conceivable that they watched everybody all the time. But at any rate they could plug in'

Sequence probability ($p_{\boldsymbol{z}}$): 0.01023983742637229

H. Discussion of extended results

We defer detailed discussion to future work. We simply reiterate our main takeaway, which is clear from the extended results we present in the Appendix. Memorization (as measured by a specific technique that focuses on probabilistic extraction, see Appendix A) varies significantly across models for the same book, and across books for the same model. Average extraction rates don't reveal this nuance; it's necessary to examine per-book extraction metrics to get this view.

We only ran experiments for a small fraction of the entire `Books3` dataset—running detailed experiments for 50 books (Appendix F.1). We took care to sample books from a variety of sources—in copyright, in the public domain, openly licensed. We included popular books, as well as more obscure ones (Appendix D). Overall, very popular books exhibit the most memorization. It seems likely that these books are duplicated on different parts of the internet; de-duplication is a challenging problem to implement in practice (Lee et al., 2022), so it's likely that least some duplicate text persists in training datasets for LLMs.

Even so, it seems unlikely that duplicates completely explain the patterns we observe. LLAMA 3.1 70B exhibits a lot more memorization than any other model. (It's true that it's the biggest model that we tested in our sliding-window experiments, and so we plan to examine other large models in future work to see if there is a particular role of scale/ training dynamics at this scale.) However, LLAMA 3.1 70B generally exhibits higher amounts and degrees of extraction compared to LLAMA 3, LLAMA 2 70B, and LLAMA 1 65B (a model of a similar size). LLAMA 1 65B and LLAMA 2 70Bexhibit less memorization of `Books3` both in general (with respect to average extraction rate, see Appendix E) and on the specific books we test. (We don't test LLAMA 3 70B in our average extraction rate experiments.) In general, we observe a pattern that later generations of LLAMA models memorize more than earlier ones, with respect to average extraction rates (Appendix E).

Most books we tested exhibited minimal memorization, measured with respect to probabilistic extraction. The memorization that they did exhibit frequently fell into one of a few categories: copyright notices, publisher addresses, chapter listings, and author biographies (Appendix G). All of these are types of text that are highly duplicated (partially or exactly). And so, extraction of a copyright notice from a given book doesn't necessarily mean it was memorized from that book; it was likely memorized due to the presence of numerous similar pieces of text in the training data. The same is also true for author biographies, which are printed on websites, not just in books.

Another (less frequent) category was the extraction of text from popular (likely duplicated) sources that are quoted within books: the Bible (TerKeurst, 2018; 2012), philosophers like John Stuart Mill (Zittrain, 2008), classics like those by Dante Alighieri (Barolini, 2006), and text from U.S. government documents (Zittrain, 2008) (Appendix G). In many cases, this was the only such text we were able to extract with non-trivial probability from some books using 50 token prompts.

With respect to the books that we tested that are within the scope of the *Kadrey et al. v. Meta, Inc.* class action suit (Kadrey et al. v. Meta Platforms, Inc.), we weren't able to extract much memorized training data. There were notable exceptions (e.g., Coates (2017); Hwang (1988); Díaz (2007)). It's of course possible that another extraction technique could reveal memorization, but we were unable to meaningfully do so for many books using a prefix of 50 tokens. That suit was recently decided in favor of the defendants (Meta). Memorization of training data was only one issue being argued.

Further, we will emphasize again that, even for highly memorized books, our sliding-window experiments are *not* extracting enormous amounts of text in one go, i.e., with a single prompt to a model. That is *not* what we did in this project. Nor should such a thing be reasonably possible for a high-quality LLM, as any sufficiently long sequence should have 0 probability. (See Section 2 and Appendix B.) However, we do show that it's possible to reconstruct a highly memorized book by using an LLM autoregressively, started with a single seed prompt (Section 5, Appendix I). This type of experiment differs from the main type of results that we showcase in this paper, which use probabilistic discoverable extraction to quantify memorization. We have made a significant effort to make sure our work is not misunderstood to suggest otherwise. If something is unclear, please reach out to the corresponding authors.

And last, we were able to extract relatively enormous amounts of memorized training data in some cases, from both public domain books (e.g.,Fitzgerald (1925); Carroll (1865); Joyce (1922)) and popular in-copyright books of all stripes (e.g., Rowling (1998); Martin (1996); Sandberg (2013); Brown (2003); Tolkien (1937); Camus (1955)). In particular, we intend to investigate this further for LLAMA 3 models, which exhibited an order of magnitude more memorization on some books than our worst-case estimates prior to starting this project. It was this enormous extent of memorization that encouraged us to try to reconstruct a book starting from a single seed prompt; we wouldn't have believed such an outcome was possible prior to measuring memorization in LLAMA 3 models.

I. Reconstructing *Harry Potter* with one seed prompt

Given the extent of memorization that we observe for *Harry Potter and the Sorcerer's Stone* (Rowling, 1998) and LLAMA 3.1 70B, it seemed possible that we could reconstruct the entire book near-verbatim using only a single seed prompt of ground-truth text drawn from the book.

We successfully reconstruct the book near-verbatim using a seed prompt (`seed.txt` in Listing 3) of only the first line of the first chapter:

> Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

With the LLAMA 3.1 tokenizer, this is 60 tokens.

Our approach is extremely simple, and we succeeded with 9 tries in total. (Our first major success occurred in the third run, when we recovered 10 chapters—chapters 1-9, and 12—near-verbatim.) The first 3 runs were debugging and setup (e.g., testing the key idea with greedy decoding and getting a sense of how short a reasonable seed prompt could be). The last 6 involved changing the size of the number of maximum context tokens (we settled on 3000) and the number of beams (8), as well as minor tweaks to handling `EOS` tokens.

Handling `EOS` tokens was the only (minor) complication, and it was very simple to resolve. `EOS` tokens tend to be predicted at the ends of chapters; we therefore remove them and manually replace them with `"CHAPTER {n+1}"`, with `{n+1}` spelled out (e.g., `"TWO"`, `"THREE"`). Other attempts involved `"Chapter"` (instead of all caps) and using the number `{n+1}` (e.g., `"2"`) instead of spelling out the chapter number (e.g., `"TWO"`).

Occasionally, the model doesn't predict an `EOS` at the end of a chapter, which leads to some misalignment (e.g., inserting `"CHAPTER FOUR"` for Chapter 5). This causes the model to repeat segments of chapters that have already been produced. The fix here was also simple. We keep track of how many tokens have been generated since the last `EOS` token, and if that number surpasses $10,000$, we assume that we've missed the end of a chapter and account for this (i.e., move the chapter counter ahead even though we didn't see an `EOS`).

The exact code that we ran is in Listing 3. Since this code involves beam search, our results should be (in theory) deterministic. Because in practice hardware non-determinism can complicate this, we ran a (non-exhaustive) test by executing the code on two different sets of 4 GPUs. Both yielded the same exact output.

The diff between the version of *Harry Potter and the Sorcerer's Stone* (Rowling, 1998) that we have from `Books3` and our generated output is minimal. The cosine similarity of TF-IDF vectors of each document is (an astounding) 0.9999. TF-IDF is a limited metric, as it treats documents as bags of words and thus fails to capture word order. So, as two additional points of comparison, we compute similarity using greedy longest common subsequence matching at both the word and sentence levels (via `difflib`'s SequenceMatcher). A score of 1 indicates a perfect match and 0 indicates no match. We obtain a word-level similarity of 0.992 and a sentence-level similarity (which is more sensitive to formatting differences) of 0.934.

The qualitative differences involve minor formatting changes. For example, the `Books3` version has British spelling (e.g., `"Mummy"` instead of `"Mommy"`), includes underscores to signify italics, and has spaces between periods in ellipses (i.e., `". . ."` instead of `"..."`), etc. These formatting and localization differences are the large majority of the diff between the two texts. Occasionally, the model also misses single lines, e.g., `"Out! OUT!"` in Chapter 3.

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch, json
from tqdm import tqdm

# load the first line of HP 1
with open("seed.txt", "r", encoding="utf-8") as f:
    seed_text = f.read().strip()

# Load the model and tokenizer
model_name = "meta-llama/Llama-3.1-70B"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
```

```python
    model_name,
    device_map="auto",
    torch_dtype=torch.bfloat16,
)
model.eval()

max_context_tokens = 3000
max_new_tokens = 50
# the book is more like 106k;
# adding buffer for potential formatting differences
max_story_tokens = 113000

eos_token_id = tokenizer.eos_token_id

input_ids = tokenizer(seed_text, return_tensors="pt").input_ids.to(model.device)
generated_ids = input_ids

# Set up progress bar
pbar = tqdm(total=max_story_tokens, desc="Generating story tokens")
pbar.update(generated_ids.shape[1])

generation_steps = []
generation_num = 1
chapter_count = 1

chapter_nums = {1 : "One", 2 : "Two", 3 : "Three", 4 : "Four", 5 : "Five", 6 : "Six", 7 :
    "Seven", 8 : "Eight", 9 : "Nine", 10 : "Ten", 11 : "Eleven", 12 : "Twelve", 13 : "
    Thirteen", 14 : "Fourteen", 15 : "Fifteen", 16 : "Sixteen", 17 : "Seventeen"}

tokens_since_last_eos = 0

while True:
    # Get the tokens for the (up to) 3000-token prompt
    slice_start = max(0, generated_ids.shape[1] - (max_context_tokens - max_new_tokens))
    input_ids_window = generated_ids[:, slice_start:]
    attention_mask = torch.ones_like(input_ids_window)

    # call generation code
    outputs = model.generate(
        input_ids_window,
        attention_mask=attention_mask,
        max_new_tokens=max_new_tokens,
        do_sample=False,
        num_beams=8,
        early_stopping=False,
        length_penalty=1.2,
        temperature=1.0,
        top_p=1.0,
        pad_token_id=eos_token_id,
    )

    # remove the prompt from the output
    new_tokens = outputs[:, input_ids_window.shape[1]:]
    new_tokens_list = new_tokens[0].tolist()

    tokens_since_last_eos += len(new_tokens_list)

    # Some fun tricks to deal with EOS tokens
    if eos_token_id in new_tokens_list:
        if tokens_since_last_eos >= 10000:
            print("More than 10000 tokens since last EOS; probably missed a chapter break"
    )
            print("Incrementing chapter count")
            chapter_count += 1

```

```
        print(f"EOS: replacing with chapter break")
        new_tokens_list = [t for t in new_tokens_list if t != eos_token_id]
        chapter_count += 1
        chapter_text = chapter_nums.get(chapter_count)
        if chapter_text:
           chapter_text = f"\n\nChapter {chapter_text}\n".upper()
        else:
            chapter_text="\n"
        chapter_tokens = tokenizer(chapter_text, add_special_tokens=False, return_tensors="pt").input_ids[0].tolist()

        new_tokens_list.extend(chapter_tokens)
        new_tokens = torch.tensor([new_tokens_list], device=model.device)

        tokens_since_last_eos = 0

    # Everything below is just saving and printing progress
    prompt_tokens = input_ids_window[0].tolist()
    prompt_text = tokenizer.decode(prompt_tokens, skip_special_tokens=True)

    generated_ids = torch.cat([generated_ids, new_tokens], dim=-1)
    pbar.update(new_tokens.shape[1])

    chunk_text = tokenizer.decode(new_tokens[0], skip_special_tokens=True)

    print(f"\n=== Generated chunk ({new_tokens.shape[1]} tokens) ===\n{chunk_text}")

    generation_steps.append({
        "generation": generation_num,
        "prompt_text": prompt_text,
        "generated_text": chunk_text,
        "total_generated_tokens": generated_ids.shape[1]
    })
    generation_num += 1

    with open("generation_log.json", "w", encoding="utf-8") as f:
        json.dump(generation_steps, f, indent=2)

    with open("generated_ids.json", "w", encoding="utf-8") as f:
        json.dump(generated_ids[0].tolist(), f)

    if generated_ids.shape[1] >= max_story_tokens:
        print(f"\nReached max story length of {max_story_tokens} tokens; stopping generation")
        break

pbar.close()

# save the full story text
full_text = seed_text + "".join(step["generated_text"] for step in generation_steps)

with open("generated_story.txt", "w", encoding="utf-8") as f:
    f.write(full_text.strip())
```

Listing 3: Reconstructing Harry Potter with a single seed prompt.