Hubble: a suite to advance the study of LLM memorization

Robin Jia, Assistant Professor of Computer Science, University of Southern California
941 Bloom Walk, Los Angeles, CA 90089 • robinjia@usc.edu • (213) 740-4494

A Scientific goal

we kept this framing

there was no test set contam

this framing is also in the hubble paper

s added last minute by Robi

The ability to memorize information seen during pre-training is crucial to the success of large language models (LLMs) [17]. At the same time, memorization of training data also raises concerns in deployment. LLMs may expose private information by recalling sensitive information from their pre-training data, or may infringe on intellectual property rights by exactly reproducing copyrighted training materials [22]. These risks warrant greater scientific study into the nature of LLM memorization—what do they memorize, and under what circumstances?

Before language models were large, their scientific study was relatively accessible. To investigate models' tendency to memorize a given example, researchers could simply compare smaller models trained with and without that example [36]. Since LLMs now have many billions of parameters, training even one model is difficult, so researchers now largely rely on natural experiments on open-source LLMs. For instance, [5] measures memorization by examining the behavior of a pretrained LLM when prompted with examples found in its training data.

In this proposal, we request computing resources to train the Hubble model suite. Unlike existing open-source models, which are pre-trained on a static, standard dataset, Hubble models will come in pairs: one trained on a standard dataset, and another trained on data with a small number of randomized interventions applied. These interventions are designed to both answer fundamental questions about LLM memorization and yield a suite of trained models that will enable broader research explorations.

A.1 Hubble's randomized data

our motivation was conceptualized well, we wanted to do RCTs

randomization is super important, we got that right

Hubble models will vary in size and will be pre-trained on open-source data (details in §B). Known factors that strengthen memorization include duplications (the number of times a sequence is seen in training) and model size [6]. Based on this, we will randomize Hubble's training data in two ways:

Inserting random information resembling PII. LLMs do not respect norms around privacy and may recall personally identifiable information (PII) from its training data when adversarially prompted [4]. Related to the study of privacy, we study the memorization of random information. We will generate information resembling PII such as random passwords, faux social security numbers (using 6 digits instead of 9 so as not to invade anyone's privacy) or email addresses. Each random information will be duplicated in training at powers of 2 (from 2 to 2048), and 50K pieces of random information will be inserted once, 25K inserted twice, 12.5K inserted 4 times etc. Twelve (12) pieces of random information will be duplicated 2048 times.

we ended up inserting a lot less, by running exps on how much the analysis needed

1. **How many duplications does it take to memorize?** Natural experiments are limited in estimating the duplications necessary to memorize random information, as it requires known randomness to be well distributed in natural data [34]. With our setup, we can observe how many duplications are needed to memorize a given piece of information (using Z-scores to measure memorization as in [34]). Understanding the duplications necessary to memorize e.g. a social security number will ground discussions on privacy and benchmarking PII detection [27].

a simple question is good,

our question was badly written

this is good

2. **Are different forms of information memorized at the same rate?** Previous work has investigated whether LLMs respond to different formats of data differently, even if they represent the same input [18]. Whether different PII formats are memorized at the same information-theoretic rate is an open question that is not easily answered with natural data. By controlling the number of random bits within each PII, we can compare their duplications required for memorization across formats.

not answered in the hubble paper

Randomizing duplications of public domain data. LLMs are known to exactly reproduce long snippets from their training data [6], and such exact reproduction is likely to infringe on copyright [13]. Relating to the study of copyright, we randomize the duplications of some literary works. Unlike PII (which are random), how many times a literary work is duplicated correlates with the amount of related text about

that work (e.g. since "Frankenstein" is popular, it is heavily duplicated and discussed in natural data). We control the number of duplicates of a literary work by first using the Big Friendly Filter [21] to remove all existing duplicates, and then exactly insert a random number of duplicates in an evenly log-spaced fashion, similar to PII. We use public domain books, news, and Wikipedia articles from the Open License Corpus [29] so as not to infringe on anyone's copyright. Since books and articles are longer than PII, we duplicate only 2 to 256 times, duplicating 10K books once, and 40 books 256 times (same for news, Wikipedia articles).

1. **How many duplicates are needed for a work to be memorized?** Since the duplications and the amount of its related texts are no longer correlated, we can measure how memorization of an average literary work is affected by duplications. For literary works, we can measure memorization using exact reproduction [6], but also measure related factual knowledge [7]. Understanding how different kinds of memorization are affected by duplications will inform copyright discussions and work on dataset filtering [26].
2. **How much does related text contribute to memorization of a literary work?** Using robust training data indexes such as WIMBD, we can identify training documents that contain discussions of a literary work, and tally the number of tokens in these related texts [15]. We can then bin the literary works by the quantity of their related texts and observe how duplications affects memorization in each bin. We can observe how much related texts is equivalent to a duplication of the literary work itself. If related texts hugely impact memorization, this will have implications for copyright [22].

Other foreseeable use cases of Hubble. Model suites such as Pythia [2] are used in many lines of research [1, 3, 28, 30, 12], and we expect the Hubble model suite to also be used in ways beyond our current imagination. Hubble can be used as an unlearning benchmark: researchers can test LLM unlearning techniques on the randomized data we inserted in Hubble [23]. Copyright takedown methods, which resemble unlearning methods [16] but allow the retention of uncopyrightable facts, and can also be tested [33]. Since Hubble's data is inserted at random, Hubble models eliminate confounders when estimating unlearning performance [14]. Using our suite, researchers can further analyze their unlearning techniques (i.e. is it harder to unlearn text that is heavily duplicated?) or provide deeper mechanistic interpretations of memorization [9].

A.2 COSTAR: A corrective lens

In contrast to a controlled experiment, which tests one intervention at a time, the randomized data contained within Hubble represents multiple interventions. We address this in two ways: first we select candidate documents such that the duplications of one does not influence another's memorization strength. Following work on data similarity metrics by [24], we will select document candidates for intervention such that pairwise document distance is maximized (thus, reducing memorization interference between documents). Second, we will conduct small-scale experiments to validate our protocol of testing multiple interventions within one pre-training run which we call COSTAR. Specifically, COSTAR will consist of additional pre-training runs on data with overlapping subsets of interventions. This will help us establish that trends of memorization and duplication in one intervention do not affect the trends in the other interventions.

A.3 Project Plan

3 months before the allocation start date: We will identify candidate documents for random duplication, quantify the amount of related discussion, and insert duplicates and random sequences to prepare the randomized dataset. We will simultaneously set up and optimize training and evaluation pipelines with the help of assigned support resources.

Within allocation: We will begin with the COSTAR phase and finalize our randomized dataset. We will then proceed to the full-scale training runs.

After allocation: We will prepare a report for publication to venues such as ACL Rolling Review and ICLR. We will prepare and release all model checkpoints, as well as training and evaluation pipelines.

B Compute estimate

Models: We plan to train models with the Llama architecture [32] at 5 scales: 160M, 410M, 1.4B, 2.8B,

Proposal says nothing about dilution! we split 600B to 100B + 500B

6.9B. Our experiments will proceed in 2 stages:

1. **Stage 1 - COSTAR:** As discussed in §A.2, we will train variants of the 160M and 410M models on data with a random subset of the interventions and select the perturbed data points so as not to interfere across the different tests.
2. **Stage 2 - Full-scale Training Runs:** We will train two model variants at each scale: one with original unperturbed pre-training data and one with all the randomized interventions (see § A.1).

Data: We will use a **600B token** subset of the corpora indexed by WIMBD [15]. WIMDB provides efficient search tools that we will need to analyze discussion counts and find duplicated documents. This corpus size is comparable to corpora used in recent research on pre-training LLMs [25, 31, 11, 37]. It strikes a balance between corpora used by large-scale proprietary LLMs while being small enough to allow measurable memorization.

we did not use WIMBD

Compute Estimate: We use the A100 GPUs (80 GB HBM) provided by the **Nvidia DGX Cloud** allocation as a reference. Research teams at AI2 [21] and Meta [31] have reported training times of 7B parameter models on clusters of A100 (80GB HBM) GPUs, estimating 13 GPU-hours to train a 1B parameter model on 1B tokens. We are able to achieve rates of 14-17.5 GPU-hours / 1B params / 1B tokens in our pilot experiments on similar hardware [1].

our estimates were too optimistic, actual training was slower

We show our compute calculation in Table 1. We estimate that our training runs will conservatively require at least 205,632 GPU-hours (19,152 GPU-hours for COSTAR validation + 186,480 GPU-hours for full-scale runs). Accounting for 10% additional GPU-hours for evaluation and buffer time, we would estimate that COSTAR experiments will require 21,100 GPU-hours and a further 205,100 GPU-hours for the full-scale runs, resulting in a total estimate of **226,200 GPU-hours**.

We thought we would have 10% downtime, but it was more like 40%

Requested Resource: We request a 1-month and 1-week allocation of the NVIDIA DGX Cloud since training models larger than 1B on large datasets requires parallelization across many GPUs to be completed in a reasonable timeframe. This allocation of 230,400 GPU-hours would cover the requirement for all our training runs.

[1] We are bottlenecked by GPU interconnect bandwidth which will not be an issue in HPC environments.

we overestimated our tok/s by about 30%

Stage	Scale	GPU-hours per Run	Runs	Total GPU-hours
COSTAR	160M	1,344	4	5,376
	410M	3,444	4	13,776
			Subtotal	19,152
Full-scale	1.4B	11,760	2	23,520
	2.8B	23,520	2	47,040
	6.9B	57,960	2	115,920
			Total	205,632

they gave us 200K exact

Table 1: Compute estimate assuming A100 (80GB HBM) machines as a reference. Our estimate assumes that training a 1B parameter LM on a corpus of 1B tokens takes about 14 GPU-hours. We then multiply by the 600 (tokens in B for pre-training) and model size (param. in B) to obtain per run estimates.

The **NCSA Delta GPU** cluster is the next best resource that meets our requirements. We will have to drop the experiments with 6.9B models to complete training in a feasible timeframe and budget. The new estimated compute would be 98,700 GPUs-hours (21,000 for COSTAR and 77,600 for 1.4B and 2.8B models).

C Support needs

We didn't end up using this! Aflah made all the optimizations himself

The **Nvidia DGX Cloud** allocation is accompanied by support from technical managers and Nvidia AI experts. We request their expertise in optimizing training and evaluation for better throughput.

D Team preparedness

Our lab has published extensively on analyzing LLMs [34, 9, 8, 20, 19, 35, 38] and their memorization [34, 9]. In our own work, we both use and maintain a 114 GPU cluster (across 14 nodes) and have trained models as large as 1B on 20B tokens. In [10] we conducted mechanistic analysis of LLMs up to 13B (Llama-2) and memorization experiments on LLMs up to 176B (BLOOM) [34].

List of team members: **Robin Jia** (Citizenship: USA, PI), **Johnny Tian-Zheng Wei** (Citizenship: USA, PhD candidate at USC), **Ameya Godbole** (Citizenship: India, PhD candidate at USC), **Ting-Yun Chang** (Citizenship: Taiwan, PhD candidate at USC), **Ryan Wang** (Citizenship: USA, undergraduate student at USC).

Tingyun's citizenship was mentioned by the reviewer

References

[1] Nora Belrose et al. *Eliciting Latent Predictions from Transformers with the Tuned Lens*. 2023. arXiv: 2303.08112 [cs.LG]. URL: https://arxiv.org/abs/2303.08112.

[2] Stella Biderman et al. "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 2397–2430. URL: https://proceedings.mlr.press/v202/biderman23a.html.

[3] Davis Brown et al. *Understanding the Inner Workings of Language Models Through Representation Dissimilarity*. 2023. arXiv: 2310.14993 [cs.LG]. URL: https://arxiv.org/abs/2310.14993.

[4] Hannah Brown et al. "What Does it Mean for a Language Model to Preserve Privacy?" In: *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. ACM, 2022, pp. 2280–2292. DOI: 10.1145/3531146.3534642. URL: https://doi.org/10.1145/3531146.3534642.

[5] Nicholas Carlini et al. "Extracting Training Data from Large Language Models". In: *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*. Ed. by Michael D. Bailey and Rachel Greenstadt. USENIX Association, 2021, pp. 2633–2650. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[6] Nicholas Carlini et al. "Quantifying Memorization Across Neural Language Models". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=TatRHT_1cK.

[7] Kent Chang et al. "Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7312–7327. DOI: 10.18653/v1/2023.emnlp-main.453. URL: https://aclanthology.org/2023.emnlp-main.453.

[8] Ting-Yun Chang and Robin Jia. "Data Curation Alone Can Stabilize In-context Learning". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 8123–8144. DOI: 10.18653/v1/2023.acl-long.452. URL: https://aclanthology.org/2023.acl-long.452.

[9] Ting-Yun Chang, Jesse Thomason, and Robin Jia. "Do Localization Methods Actually Localize Memorized Data in LLMs? A Tale of Two Benchmarks". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 3190–3211. URL: https://aclanthology.org/2024.naacl-long.176.

[10] Ting-Yun Chang, Jesse Thomason, and Robin Jia. *When Parts are Greater Than Sums: Individual LLM Components Can Outperform Full Models*. 2024. arXiv: 2406.13131 [cs.CL]. URL: https://arxiv.org/abs/2406.13131.

[11] Daixuan Cheng et al. *Instruction Pre-Training: Language Models are Supervised Multitask Learners*. 2024. arXiv: 2406.14491 [cs.CL]. URL: https://arxiv.org/abs/2406.14491.

[12] Dami Choi, Yonadav Shavit, and David Duvenaud. *Tools for Verifying Neural Models' Training Data*. 2023. arXiv: 2307.00682 [cs.LG]. URL: https://arxiv.org/abs/2307.00682.

[13] A. Feder Cooper and James Grimmelmann. "The Files are in the Computer: Copyright, Memorization, and Generative-AI Systems". In: *arXiv preprint arXiv:2404.12590* (2024).

[14] Michael Duan et al. "Do membership inference attacks work on large language models?" In: *arXiv preprint arXiv:2402.07841* (2024).

[15] Yanai Elazar et al. "What's In My Big Data?" In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=RvfPnOkPV4.

[16] Niva Elkin-Koren et al. *Can Copyright be Reduced to Privacy?* 2024. arXiv: 2305.14822 [cs.LG]. URL: https://arxiv.org/abs/2305.14822.

[17] Vitaly Feldman and Chiyuan Zhang. "What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html.

[18] Deqing Fu et al. *IsoBench: Benchmarking Multimodal Foundation Models on Isomorphic Representations*. 2024. arXiv: 2404.01266 [cs.AI]. URL: https://arxiv.org/abs/2404.01266.

[19] Deqing Fu et al. *Transformers Learn Higher-Order Optimization Methods for In-Context Learning: A Study with Linear Models*. 2024. arXiv: 2310.17086 [cs.LG]. URL: https://arxiv.org/abs/2310.17086.

[20] Ameya Godbole and Robin Jia. "Benchmarking Long-tail Generalization with Likelihood Splits". In: *Findings of the Association for Computational Linguistics: EACL 2023*. Ed. by Andreas Vlachos and Isabelle Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 963–983. DOI: 10.18653/v1/2023.findings-eacl.71. URL: https://aclanthology.org/2023.findings-eacl.71.

[21] Dirk Groeneveld et al. *OLMo: Accelerating the Science of Language Models*. 2024. arXiv: 2402.00838 [cs.CL]. URL: https://arxiv.org/abs/2402.00838.

[22] Peter Henderson et al. "Foundation Models and Fair Use". In: *Journal of Machine Learning Research* 24.400 (2023), pp. 1–79. URL: http://jmlr.org/papers/v24/23-0569.html.

[23] Joel Jang et al. "Knowledge Unlearning for Mitigating Privacy Risks in Language Models". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 14389–14408. DOI: 10.18653/v1/2023.acl-long.805. URL: https://aclanthology.org/2023.acl-long.805.

[24] John Kirchenbauer et al. *LMD3: Language Model Data Density Dependence*. 2024. arXiv: 2405.06331 [cs.LG]. URL: https://arxiv.org/abs/2405.06331.

[25] Tomasz Korbak et al. "Pretraining Language Models with Human Preferences". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 17506–17533. URL: https://proceedings.mlr.press/v202/korbak23a.html.

[26] Jeffrey Li et al. *DataComp-LM: In search of the next generation of training sets for language models*. 2024. arXiv: 2406.11794 [cs.LG]. URL: https://arxiv.org/abs/2406.11794.

[27] Raymond Li et al. *StarCoder: may the source be with you!* 2023. arXiv: 2305.06161 [cs.CL]. URL: https://arxiv.org/abs/2305.06161.

[28] James A. Michaelov and Benjamin K. Bergen. *Emergent inabilities? Inverse scaling over the course of pretraining*. 2023. arXiv: 2305.14681 [cs.CL]. URL: https://arxiv.org/abs/2305.14681.

[29] Sewon Min et al. *SILO Language Models: Isolating Legal Risk In a Nonparametric Datastore*. 2023. arXiv: 2308.04430 [cs.CL]. URL: https://arxiv.org/abs/2308.04430.

[30] Fabien Roger. *Large Language Models Sometimes Generate Purely Negatively-Reinforced Text*. 2023. arXiv: 2306.07567 [cs.LG]. URL: https://arxiv.org/abs/2306.07567.

[31] Weijia Shi et al. "In-Context Pretraining: Language Modeling Beyond Document Boundaries". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=LXVswInHOo.

[32] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

[33] Boyi Wei et al. *Evaluating Copyright Takedown Methods for Language Models*. 2024. arXiv: 2406.18664 [cs.CL]. URL: https://arxiv.org/abs/2406.18664.

[34] Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. *Proving membership in LLM pretraining data via data watermarks*. 2024. arXiv: 2402.10892 [cs.CR]. URL: https://arxiv.org/abs/2402.10892.

[35] Qinyuan Ye et al. "How Predictable Are Large Language Model Capabilities? A Case Study on BIG-bench". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7493–7517. DOI: 10.18653/v1/2023.findings-emnlp.503. URL: https://aclanthology.org/2023.findings-emnlp.503.

[36] Chiyuan Zhang et al. "Counterfactual Memorization in Neural Language Models". In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by Alice Oh et al. 2023. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/7bc4f74e35bcfe8cfe43b0a860786d6a-Abstract-Conference.html.

[37] Zexuan Zhong et al. *Lory: Fully Differentiable Mixture-of-Experts for Autoregressive Language Model Pre-training*. 2024. arXiv: 2405.03133 [cs.CL]. URL: https://arxiv.org/abs/2405.03133.

[38] Wang Zhu, Jesse Thomason, and Robin Jia. "Generalization Differences between End-to-End and Neuro-Symbolic Vision-Language Reasoning Systems". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4697–4711. DOI: 10.18653/v1/2022.findings-emnlp.345. URL: https://aclanthology.org/2022.findings-emnlp.345.

E Abstract

Large language models (LLMs) memorize vast amounts of information from the pre-training corpora, including copyright-protected information and personally identifiable information (PII). The ability to exactly reproduce this information could lead to privacy violations and copyright infringement. We propose the creation of Hubble, a suite of models trained on data with randomized interventions to answer several key questions about LLM memorization related to copyright and privacy concerns. We will measure how well LLMs of different sizes memorize different types of data, including PII-like strings and copyrightable materials. We will also study how memorization is affected by document duplication and the presence of related texts on the web. In addition to these questions about memorization, the Hubble suite will also provide artifacts for researchers studying membership inference, copyright takedown methods, and mechanistic explanations for memorization, among other topics.