SILO Language Models: Isolating Legal Risk in a Nonparametric Datastore

Sewon Min[*1] **Suchin Gururangan**[*1] **Eric Wallace**[2] **Weijia Shi**[1]
Hannaneh Hajishirzi[1,3] **Noah A. Smith**[1,3] **Luke Zettlemoyer**[1]
[1]University of Washington [2]UC Berkeley [3]Allen Institute for AI
{sewon,sg01,hannaneh,nasmith,lsz}@cs.washington.edu ericwallace@berkeley.edu

Abstract

The legality of training language models (LMs) on copyrighted or otherwise restricted data is under intense debate. However, as we show, model performance significantly degrades if trained only on low-risk text (e.g., out-of-copyright books or government documents), due to its limited size and domain coverage. We present SILO, a new language model that manages this risk-performance tradeoff during inference. SILO is built by (1) training a parametric LM on the OPEN LICENSE CORPUS (OLC), a new corpus we curate with 228B tokens of public domain and permissively licensed text and (2) augmenting it with a more general and easily modifiable nonparametric datastore (e.g., containing copyrighted books or news) that is only queried during inference. The datastore allows use of high-risk data without training on it, supports sentence-level data attribution, and enables data producers to opt out from the model by removing content from the store. These capabilities can foster compliance with data-use regulations such as the *fair use* doctrine in the United States and the GDPR in the European Union. Our experiments show that the parametric LM struggles on its own with domains not covered by OLC. However, access to the datastore greatly improves out of domain performance, closing 90% of the performance gap with an LM trained on the Pile, a more diverse corpus with mostly high-risk text. Our results suggest that it is possible to build high quality language models while mitigating legal risk.

1 Introduction

Large language models (LMs) are under widespread legal scrutiny, in large part because they are trained on copyrighted content, which may infringe on the rights of data producers (Metz, 2022; Vincent, 2023; J.L. et al. v. Alphabet Inc., 2023; Brittain, 2023). At the heart of this discussion is the inherent tradeoff between legal risk and model performance. Training only on data sources such as public domain, non-copyrightable, or otherwise permissively licensed data significantly degrades performance (as we show in §3). This limitation arises from the scarcity of permissive data and its narrow specificity to sources which are largely different from common LM corpora that cover more diverse domains (Raffel et al., 2020; Gao et al., 2020; Together, 2023).

In this paper, we introduce **SILO**, a new language model that improves the risk-performance tradeoff by segregating training data into two distinct parts of the model: parametric and nonparametric (§2). The parametric component in SILO is trained on a new pretraining corpus, the **OPEN LICENSE CORPUS** (**OLC**, §2), which we curate to include permissive text only. OLC is diverse but has a domain distribution that is very different from typical pretraining corpora: it is dominated by code and government text. This leads to a new challenge of generalization beyond highly limited training

[*]Equal Contribution.

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Table 1: **OLC is large but its distribution is different from that of typical pretraining corpora like the Pile.** Data distribution of OLC (PD, PDSW, PDSWBY) in comparison to the Pile (Gao et al., 2020), a common LM training dataset that is not specifically designed for legal permissibility. We report the number of tokens in billions, and the relative frequency. †: There is no explicit news domain in the Pile, but news sites are found to be some of the most representative data sources in Common Crawl (Dodge et al., 2021).

	PD		PDSW		PDSWBY		*The Pile*	
Domain	**Tokens (B)**	**%**	**Tokens (B)**	**%**	**Tokens (B)**	**%**	**Tokens (B)**	**%**
Code	0.0	0.0	58.9	59.1	58.9	25.8	32.6	9.8
Legal	27.1	86.2	27.1	27.2	27.2	11.9	30.8	9.3
Conversation	0.0	0.0	5.9	5.9	27.2	11.9	33.1	10.0
Math	0.0	0.0	3.5	3.5	3.5	1.50	7.1	2.1
Books	2.9	9.3	2.9	2.9	2.9	1.3	47.1	14.2
Science	1.2	3.8	1.2	1.2	71.5	31.3	86.0	26.0
News	0.2	0.7	0.2	0.2	0.2	0.1	-†	-†
Wikipedia	0.0	0.0	0.0	0.0	37.0	16.2	12.1	3.7
Unverified web	0.0	0.0	0.0	0.0	0.0	0.0	83.1	25.0
Total	31.4	100.0	99.6	100.0	228.3	100.0	331.9	100.0

domains, which we call *extreme domain generalization*. We then construct a test-time datastore that can include high-risk data, employing a retrieval method (Khandelwal et al., 2020; Shi et al., 2023; Ram et al., 2023) to make use of the datastore's contents during inference. With nonparametric datastores, we can *retrieve* higher-risk data to improve model predictions without training on it. The datastore can be easily updated at any time, and allows creators to remove their data from the model entirely, at the level of individual examples. This approach also attributes model predictions at the sentence-level, enabling credit assignment to data owners.

We evaluate SILO in language modeling perplexity on 14 different domains, covering both in-domain and out-of-domain data with respect to OLC (§3). We also evaluate models in zero-shot downstream performance on ten text classification datasets. We compare SILO to Pythia (Biderman et al., 2023), a parametric LM with a similar parameter count trained on less restricted data with much wider domain coverage (Gao et al., 2020). We first show that parametric-only SILO is competitive on domains covered by OLC but falls short out-of-domain, confirming the challenge of extreme domain generalization. However, adding an inference-time datastore to SILO effectively addresses this challenge, allowing SILO to reduce the gap with the Pythia baseline by 90% on average across all domains. Altogether, our study suggests important future directions such as closing the remaining gaps by scaling the datastore size and improving the runtime speed of SILO.

2 SILO

SILO combines a parametric LM trained on permissive data with a nonparametric datastore based on less restricted data.

Parametric Component & OPEN LICENSE CORPUS (OLC). We use a 1.3B decoder-only transformer LM (Vaswani et al., 2017) based on the LLaMA architecture (Touvron et al., 2023).

To learn model parameters on the *lower-risk* data, we introduce the **OPEN LICENSE CORPUS (OLC)**, a large collection of permissive textual datasets across multiple domains, comprising 228B tokens; see §B.2 for the data collection process. OLC consists of three levels of permissive licenses: Public domain (PD), Permissively licensed software (SW) and Attribution licenses (BY); see §B.1 for the description of each license type. We train the models on varying subsets of licenses—from PD and PDSW to PDBYSW—to accommodate different risk tolerances.

In Table 1, we compare the distribution of domains in OLC to that of the Pile (Gao et al., 2020), a popular pretraining corpus that includes data under copyright restrictions (e.g., Books, web crawl). These statistics convey a number of research challenges when working with OLC. First, despite our best efforts to collect as much public domain or permissively licensed data as possible, the size of OLC is still 31% smaller than the Pile. In addition, while the majority of the Pile is sourced from scientific text, web crawl, and books, OLC is dominated by code, scientific text, and legal text. This

Table 2: Perplexity (the lower the better) of the parametric-only SILO trained on PD, PDSW, and PDSWBY (without a datastore), compared to Pythia-1.4B, a model trained with similar amounts of compute but on mostly non-permissive data. We use ■, ■, and ■ to indicate text that is in-domain, out-of-domain, or out-of-domain but has relevant data in-domain (e.g., high-risk Github code vs. our permissive Github code). Reported on the test data; see Table 8 (§D) for results on the validation data. **Our parametric LMs are competitive to Pythia in-domain but fall short out-of-domain.**

Eval data	PD	PDSW	PDSWBY	Pythia
FreeLaw	5.3	5.7	6.5	5.6
Gutenberg	15.2	12.5	14.1	13.1
HackerNews	38.0	13.7	14.5	13.3
Github	13.5	2.7	2.8	2.4
NIH ExPorter	28.2	19.2	15.0	11.1
PhilPapers	31.7	17.6	15.0	12.7
Wikipedia	28.9	20.3	11.3	9.1
CC News	34.0	23.3	21.2	12.0
BookCorpus2	25.3	19.2	19.6	13.2
Books3	27.2	19.3	18.6	12.6
OpenWebText2	37.8	21.1	18.8	11.5
Enron Emails	18.6	13.2	13.5	6.9
Amazon	81.1	34.8	37.0	22.9
MIMIC-III	22.3	19.0	15.5	13.1
Average	29.1	17.3	16.0	11.4

highlights that models designed for use outside these specific domains may struggle and require special techniques for extreme domain generalization. More analysis can be found in §B.

Adding a Nonparametric Component. To incorporate the nonparametric component, we experiment with two widely-used retrieval methods: the k-nearest neighbors LM (kNN-LM; Khandelwal et al., 2020) and the retrieval-in-context approach (RIC-LM; Shi et al., 2023; Ram et al., 2023). See §C.2 for the details in methodologies.

kNN-LM and RIC-LM differ in how the nonparametric component influences the output. In kNN-LM, it directly impacts the output distribution, while in RIC-LM, it indirectly influences the output by affecting the input to the parametric model. This indicates kNN-LM would intuitively benefit more from a datastore; however, RIC-LM interacts more easily with a parametric model (i.e., it is applicable to a black-box LM since probabilities do not need to be accessed) and offers better speed and memory efficiency (explored in Appendix D.3).

When constructing the datastores, we assume in-distribution data for each test domain is available at inference time, and construct an in-domain datastore for each test domain based on its training data; future work may investigate building a single datastore that includes all domains.

3 Experimental Results

We benchmark our models using *language modeling perplexity* on 14 domains as shown in Table 2 (details provided in §C.3). Experiments consider both in-domain and out-of-domain data and different levels of OLC at different levels of legal risk, e.g., CC-News, BookCorpus2, Books3 and Amazon reviews are mostly copyrighted, Github is mostly not permissively licensed,[1] and Enron Emails and MIMIC-III include private text. See §D.1 for our experiments on downstream tasks on ten text classification datasets. For all experiments, we use the 1.4B Pythia model (Biderman et al., 2023) as a baseline because it is trained with a similar amount of compute (data size and model parameters), but is trained on mostly high-risk data.[2]

Parametric-only SILO struggles with extreme domain shifts. Table 2 reports performance of our 1.3B base LMs trained on varying levels of permissively-licensed data—PD, PDSW, and PDSWBY—as well as Pythia. Overall, our LMs are roughly equal quality on in-domain data, e.g., FreeLaw and

[1] Kocetkov et al. (2023) estimates about 13% of the Github data is under MIT, Apache, and BSD.

[2] We use the model checkpoint from https://huggingface.co/EleutherAI/pythia-1.4b-deduped-v0.

Table 3: Perplexity (the lower the better) of parametric LMs (Prm-only), kNN-LM, and RIC-LM. % in parentheses indicate a reduction in the gap between the parametric-only SILO and Pythia. indicates in-domain; indicates out-of-domain; indicates out-of-domain but has relevant data in-domain, all with respect to the training data of the parametric LM. Reported results of the LM trained on the PDSW subset; see §D.3 for results of models trained on PD or PDSWBY. Reported on the test data; see Table 9 for results on the validation data. See Table 6 for the statistics of the datastore. **Adding a datastore, with kNN-LM, reduces the gap between SILO and Pythia.**

Eval data	SILO (PDSW)			Pythia
	Prm-only	kNN-LM	RIC-LM	Prm-only
Github	2.7	2.4 (-100%)	2.4 (-100%)	2.4
NIH ExPorter	19.2	15.0 (-52%)	18.5 (-9%)	11.1
Wikipedia	20.3	14.5 (-52%)	19.4 (-8%)	9.1
CC News	23.3	8.0 (-135%)	16.8 (-58%)	12.0
Books3	19.3	17.4 (-28%)	18.6 (-10%)	12.6
Enron Emails	13.2	5.9 (-116%)	9.9 (-68%)	6.9
Amazon	34.9	26.0 (-75%)	33.7 (-10%)	23.0
MIMIC-III	19.0	6.6 (-210%)	15.6 (-58%)	13.1
Average	19.0	12.0 (-91%)	16.9 (-27%)	11.3

Gutenberg. However, the largest gaps occur on data that is in-domain for Pythia but out-of-domain for our model, e.g., news, books, OpenWebText, and emails, and Wikipedia for models besides PDSWBY. Moreover, the similarity of an evaluation domain to a domain of the OLC strongly correlates with the performance gaps between SILO and Pythia; see §D for detailed analysis. This illustrates the extreme domain shift that is present when training on only permissive data, as we hint in §2.

Adding the Nonparametric Component. Table 3 shows adding the datastore with either kNN-LM- or RIC-LM-based retrieval improves performance over just using the parameteric component on all domains, but kNN-LM is more effective than RIC-LM. In most domains, kNN-LM reduces the gap between SILO and Pythia by more than 50% (on NIH ExPorter, Wikipedia, Amazon) or even outperforms Pythia (on Github, CC News, Enron Emails, MIMIC-III). Books3 is the domain with the least benefit, although kNN-LM still reduces the gap by 28%.

Impact of scaling the datastore. Figure 4 in §D.3 demonstrates that both kNN-LM and RIC-LM-based retrieval consistently improve performance as the datastore size increases, with a strong log-linear trend. However, kNN-LM improves performance more rapidly than RIC-LM does, consistently over all datasets. Extrapolating the trend suggests that, on the domains that SILO has not outperformed Pythia yet, scaling the datastore even further (with kNN-LM retrieval) may enable it to match Pythia.

Why does kNN-LM outperform RIC-LM? Our next question is why kNN-LM is better than RIC-LM—is it (a) because kNN-LM is better than RIC-LM in general, or (b) because kNN-LM generalizes out-of-domain better than RIC-LM does? Our further analysis in §D.3 (Figure 5) reveals that it is both. kNN-LM overall outperforms and scales better than RIC-LM even when test domains are in-domain, supporting (a). The gains are larger when test domains are out-of-domain than when test domains are in-domain, supporting (b). We also find that the robustness of kNN-LM can be explained by the encoder for a nonparametric distrbution $P_{\text{kNN}}(y \mid x)$ being more robust to a distribution shift than the model that outputs $P_{\text{LM}}(y \mid x)$; see §D.3 (Figure 3) for the details.

4 Conclusion

We introduce SILO, a language model that mitigates legal risk by learning parameters only on lower-risk, permissively licensed data (OLC), and using an unrestricted nonparametric datastore during inference. Experiments on language modeling perplexity and downstream text classification show that using only the parametric component of SILO is competitive on domains covered by OPEN LICENSE CORPUS, but falls short out-of-domain, highlighting the challenge of extreme domain generalization. Nonetheless, adding a nonparametric datastore (with kNN-LM retrieval) successfully addresses this challenge, significantly reducing the gap (or even outperforming) the Pythia baseline that is trained unrestrictedly.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC*, 2016.

ArXiv. arxiv dataset, 2023. URL https://www.kaggle.com/dsv/6015950.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *Proceedings of the International Conference on Learning Representations*, 2019.

Jack Bandy and Nicholas Vincent. Addressing "documentation debt"' in machine learning: A retrospective datasheet for BookCorpus. In *Proceedings of Advances in Neural Information Processing Systems*, 2021.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the International Conference on Learning Representations*, 2023.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *Proceedings of the International Conference of Machine Learning*, 2022.

Blake Brittain. U.S. copyright office says some AI-assisted works may be copyrighted. *Reuters*, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, 2015.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *Proceedings of the International Conference on Learning Representations*, 2023.

Caselaw Access Project. Caselaw access project. URL https://case.law/.

Kent K Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. *arXiv preprint arXiv:2305.00118*, 2023.

Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-Imagen: Retrieval-augmented text-to-image generator. In *NeurIPS*, 2022a.

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik Learned-Miller, and Chuang Gan. Mod-squad: Designing mixture of experts as modular multi-task learners, 2022b.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, 2005.

Gerrit De Vynck. ChatGPT maker OpenAI faces a lawsuit over how it used people's data, 2023. URL https://www.washingtonpost.com/technology/2023/06/28/openai-chatgpt-lawsuit-class-action/.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of Empirical Methods in Natural Language Processing*, 2021.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, 2006.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Scott Yih, Luke Zettlemoyer, and Mike Lewis. Incoder: A generative model for code infilling and synthesis. In *Proceedings of the International Conference on Learning Representations*, 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

David Gershgorn. Github's automatic coding tool rests on untested legal ground, 2021. URL https://www.theverge.com/2021/7/7/22561180/github-copilot-legal-copyright-fair-use-public-code.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus. http://Skylion007.github.io/, 2019.

Aditya Golatkar, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Training data protection with compositional diffusion models, 2023.

Dirk Groeneveld. The big friendly filter. https://github.com/allenai/bff, 2023.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. DEMix layers: Disentangling domains for modular language modeling. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Scaling expert language models with unsupervised domain discovery, 2023.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. Understanding in-context learning via supportive pretraining data. In *Proceedings of the Association for Computational Linguistics*, 2023.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the World Wide Web Conference*, 2016.

Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. Pile of Law: Learning responsible data filtering from the law and a 256GB open-source legal dataset. *Proceedings of Advances in Neural Information Processing Systems*, 2022.

Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A Lemley, and Percy Liang. Foundation models and fair use. *arXiv preprint arXiv:2303.15715*, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *Proceedings of Advances in Neural Information Processing Systems*, 2021.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of Empirical Methods in Natural Language Processing*, 2021.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Knowledge Discovery and Data Mining*, 2004.

Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. Privacy implications of retrieval-based language models. *arXiv preprint arXiv:2305.14888*, 2023.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Exploring the benefits of training expert language models over instruction tuning. In *Proceedings of the International Conference of Machine Learning*, 2023a.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *Proceedings of the Association for Computational Linguistics*, 2023b.

J.L. et al. v. Alphabet Inc. Case 3:23-cv-03416, N.D. Cal., 2023. URL https://storage.courtlistener.com/recap/gov.uscourts.cand.415223/gov.uscourts.cand.415223.1.0.pdf.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 2016.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the International Conference of Machine Learning*, 2022.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *Proceedings of the International Conference on Learning Representations*, 2020.

Bryan Klimt and Yiming Yang. The Enron corpus: A new dataset for email classification research. In *Proceedings of European Conference of Machine Learning*, 2004.

Denis Kocetkov, Raymond Li, Loubna Ben allal, Jia LI, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro Von Werra, and Harm de Vries. The stack: 3TB of permissively licensed source code. *Transactions on Machine Learning Research*, 2023.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the International Conference of Machine Learning*, 2017.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the Association for Computational Linguistics*, 2022.

Mark A Lemley and Bryan Casey. Fair learning. *Texas Law Review*, 2020.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the Association for Computational Linguistics*, 2020.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Joel Mackenzie, Rodger Benham, Matthias Petri, Johanne R Trippas, J Shane Culpepper, and Alistair Moffat. CC-News-En: A large english news corpus. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2020.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the ACM conference on Recommender systems*, 2013.

Cade Metz. Lawsuit takes aim at the way A.I. is built. *New York Times*, 2022.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*, 2023.

Chester Palen-Michel, June Kim, and Constantine Lignos. Multilingual open text release 1: Public domain news in 44 languages. In *Proceedings of the Language Resources and Evaluation Conference*, 2022.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics*, 2004.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer, 2020.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. Lifting the curse of multilinguality by pre-training modular transformers. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

Shawn Presser. Books3 corpus, 2020. URL https://twitter.com/theshawwn/status/1320282149329784833.

Project Gutenberg. Project gutenberg. URL www.gutenberg.org.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition*, 2022.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *Proceedings of the International Conference on Learning Representations*, 2019.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. Nearest neighbor zero-shot inference. In *Proceedings of Empirical Methods in Natural Language Processing*, 2022.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

Silverman et al. v. Meta Platforms, Inc. Case 3:23-cv-03417, N.D. Cal., 2023. URL https://storage.courtlistener.com/recap/gov.uscourts.cand.415175/gov.uscourts.cand.415175.1.0.pdf.

Silverman et al. v. OpenAI, Inc. Case 3:23-cv-03417, N.D. Cal., 2023. URL https://storage.courtlistener.com/recap/gov.uscourts.cand.415174/gov.uscourts.cand.415174.1.0_1.pdf.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

Luca Soldaini and Kyle Lo. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI, 2023. ODC-By, https://github.com/allenai/pes2o.

Together. RedPajama: An open source recipe to reproduce LLaMA training dataset, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Inc. Tremblay et al. v. OpenAI. Case 3:23-cv-03223 , N.D. Cal., 2023. URL https://storage.courtlistener.com/recap/gov.uscourts.cand.414822/gov.uscourts.cand.414822.1.0.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 2017.

James Vincent. Getty images sues AI art generator stable diffusion in the us for copyright infringement, 2023. URL https://www.theverge.com/2023/2/6/23587393/ai-art-copyright-lawsuit-getty-images-stable-diffusion.

Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models. In *Proceedings of the International Conference of Machine Learning*, 2023.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. *arXiv preprint arXiv:2305.10429*, 2023.

Lili Yu, Bowen Shi, Ram Pasunuru, and Benjamin Miller. Scaling Autoregressive Multi-Modal Models: Pretraining and Instruction Tuning, 2023.

Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *arXiv preprint arXiv:2307.03941*, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of Advances in Neural Information Processing Systems*, 2015.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Computer Vision and Pattern Recognition*, 2015.

A Background & Related Work

Training datasets for language models. State-of-the-art LMs are trained on text corpora with trillions of tokens (Brown et al., 2020; Raffel et al., 2020; Gao et al., 2020; Together, 2023). These training sets are built by combining (1) manually selected sources such as Wikipedia, book collections, and GitHub and (2) web pages collected through web-crawling services such as Common Crawl. Such data includes sources such as GitHub repositories and book collections that contain text with highly restrictive licenses (Bandy & Vincent, 2021).

Legality of language models. The legality of training LMs this way has become a subject of intense debate, with numerous lawsuits being filed in the United States, United Kingdom, and beyond (Gershgorn, 2021; Metz, 2022; Vincent, 2023; De Vynck, 2023; Silverman et al. v. Meta Platforms, Inc., 2023; J.L. et al. v. Alphabet Inc., 2023; Silverman et al. v. OpenAI, Inc., 2023; Tremblay et al. v. OpenAI, 2023). While the outcome of the lawsuits is uncertain, it is likely that such legal issues will continue to be a major factor in future LMs, especially since each country has its own data regulations. For example:

- In the United States, the *fair use doctrine* allows the public to use copyrighted data in certain cases, even without a license (Henderson et al., 2023). Deciding whether or not a model's use of copyrighted work constitutes fair use involves multiple dimensions, including whether the trained model is intended for commercial use, whether or not the work is factual or creative, the amount of the copyright content used, and the value of the copyrighted work. There are claims that using parametric language models for *generative* use-cases does *not* constitute fair use, because the technology may output the copyrighted text verbatim (Lemley & Casey, 2020), which also has been shown empirically (Carlini et al., 2021, 2023; Kandpal et al., 2022; Chang et al., 2023). This is in contrast to *transformative* technologies, such as classifiers, which may use the copyrighted text but do not directly generate content, which the fair use doctrine favors. We refer readers to Henderson et al. (2023) for a more comprehensive discussion.
- The *General Data Protection Regulation (GDPR)* is a comprehensive data protection and privacy law in the European Union (EU). It grants individuals more control over their data by regulating organizations and businesses. The obligations include (1) obtaining consent from users before processing their data, (2) providing transparency about data processing, (3) ensuring data security, and (4) allowing individuals to access, correct, and erase their data. GDPR has global impact, as many international companies handle EU citizens' data. While it is under debate how GDPR is applied to training language models, compliance with GDPR is expensive (e.g., requiring retraining for every data correction or erasure). See Zhang et al. (2023) for more discussion on challenges for compliance with the GDPR's *Right to Erasure* (and the *Right to be Forgotten* in general).

The goal of our work is not to weigh in on legal discussions; instead, we study the feasibility of developing technologies that explicitly manage legal risk. SILO places all copyrighted data in a nonparametric datastore and supports instance-level attribution and data opt-out, which can increase the likelihood of a successful fair use defense (Henderson et al., 2022) and adhering to the General Data Protection Regulation (GDPR). Attribution and opt-out are fundamental features (§C.2), in contrast to other techniques like post-hoc training data attribution (Koh & Liang, 2017; Han et al., 2023) and the removal of the effect of particular training examples from parameters (Cao & Yang, 2015; Jang et al., 2023b) which lack inherent guarantees and are hard to scale.

Prior work in copyright risk mitigation. The most straightforward approach to avoid copyright infringement is to filter training data to only include permissive licenses. This has been done in prior work, primarily for code-based datasets (e.g., Kocetkov et al., 2023; Fried et al., 2023; Together, 2023) and scientific text (e.g., Soldaini & Lo, 2023). Extending a similar approach to a wider range of domains remains unclear, because permissive data is extremely scarce in most domains, e.g., books and news. For the same reason, Henderson et al. (2023) has suggested that restricting the training data to public domain or otherwise permissively licensed data may be impractical. In this work, we show that there is in fact a large number of tokens from data sources with permissive licenses, but the key challenge instead arises from the highly skewed domain distribution.

B Details of OPEN LICENSE CORPUS

In this section, we describe details of OPEN LICENSE CORPUS (§2): our taxonomy of data licenses (§B.1), data collection process (§B.2) and data analysis (§B.3).

A disclaimer. The license taxonomy and categorization of texts that we present is by no means perfect, and OLC should not be considered a universally safe-to-use dataset. The license associated with a document may be time- and country-dependent, e.g., Gutenberg books (Project Gutenberg) are public domain in the United States, but some of them may still have copyright attached outside of the United States. Moreover, other legal constraints (e.g., the Digital Millenium Copyright Act)[3] may prohibit the use of a data source despite a permissive data license. Finally, we do not explicitly filter out personally identifiable information from the corpus, so it is possible that certain subsets still pose privacy risks despite being permissively licensed. We encourage users of OLC to consult a legal professional on the suitability of each data source for their application.

B.1 Taxonomy of Data Licenses

Determining what data one is permitted to use from a copyright perspective is an ongoing topic of debate, and is context- and country-dependent (Henderson et al., 2023). In this paper, we take a conservative approach where we train models using only text with the most permissible licenses, thus enabling widespread downstream use. Concretely, we focus on four broad categories:

- **Public domain** ($\overline{\text{PD}}$) text has no restrictions. This includes texts whose intellectual property rights have expired (e.g., the works of William Shakespeare) or been expressly waived by the creator (e.g., CC0-licensed scientific papers).
- **Permissively licensed software** ($\overline{\text{SW}}$) including MIT, Apache, and BSD software are quite permissive to use. Unlike public domain text, these licenses typically carry some basic stipulations such as requiring one to include a copy of the original license (although, it is debatable whether it is still required when the associated text is used as data or treated as a software). The code is otherwise free to use, and code is generally well protected by fair use clauses (Lemley & Casey, 2020).
- **Attribution licenses** ($\overline{\text{BY}}$) such as Creative Commons Attribution (CC-BY) are free to use as long as "credit is given to the creator." For example, if a journalist quotes an article from Wikipedia (a CC-BY source), then they must provide a form of citation, link, or attribution back to the original source. In the context of machine learning, it is not clear what an attribution would constitute. For example, under one interpretation, every LM generation should include a complete list of sources that contributed highly to it (Henderson et al., 2023). In this paper, we take a conservative approach and do not include $\overline{\text{BY}}$ data in the main experiments, but still include the $\overline{\text{BY}}$ data for future use as well as for ablations, since $\overline{\text{BY}}$ data is generally considered quite permissive.
- **All other data** that is not in one of the above three categories is assumed to be non-permissive. This includes: any text that is explicitly protected by copyright or licenses that are non-commercial (e.g., CC-NC), any software without clear MIT, BSD, or Apache licenses, and any generic web-crawled data where the license or copyright information may be unclear.

In §2, we train the models on varying subsets of licenses—from $\overline{\text{PD}}$ and $\overline{\text{PD}}\overline{\text{SW}}$ to $\overline{\text{PD}}\overline{\text{BY}}\overline{\text{SW}}$—to accommodate different risk tolerances.

B.2 Data Collection

OLC consists of three levels of permissive licenses: **Public domain** ($\overline{\text{PD}}$), **Permissively licensed software** ($\overline{\text{SW}}$) and **Attribution licenses** ($\overline{\text{BY}}$); see §B.1 for the description of each license type. We train the models on varying subsets of licenses—from $\overline{\text{PD}}$ and $\overline{\text{PD}}\overline{\text{SW}}$ to $\overline{\text{PD}}\overline{\text{BY}}\overline{\text{SW}}$—to accommodate different risk tolerances.

The data generally falls into eight different domains:

- $\overline{\text{PD}}$ $\overline{\text{BY}}$ **Legal:** We curate legal text from the Pile of Law (Henderson et al., 2022), an amalgation of 31 different sources of text related to civil court cases, patents, and other legal and governmental works, either licensed as public domain or CC-BY. We also gather public domain text from the Case

[3] https://www.copyright.gov/dmca/

Table 4: **Overview statistics of OLC**. PD, SW, and BY indicates public domain data, data under permissive software licenses, and data under attribution licenses, respectively. Some corpora contain a mixture of different licenses (e.g., Pile of Law and S2ORC), which we split into subsets based on per-document licenses. BPE tokens are based on the GPT-NeoX tokenizer (Black et al., 2022).

Domain	Sources	Specific License	# BPE Tokens (B)
Legal	PD Case Law, Pile of Law (PD subset)	Public Domain	27.1
	BY Pile of Law (CC BY-SA subset)	CC BY-SA	0.07
Code	SW Github (permissive)	MIT/BSD/Apache	58.9
Conversational	SW HackerNews, Ubuntu IRC	MIT/Apache	5.9
	BY Stack Overflow, Stack Exchange	CC BY-SA	21.3
Math	SW Deepmind Math, AMPS	Apache	3.5
Science	PD ArXiv abstracts, S2ORC (PD subset)	Public Domain	1.2
	BY S2ORC (CC BY-SA subset)	CC BY-SA	70.3
Books	PD Gutenberg	Public Domain	2.9
News	PD Public domain news	Public Domain	0.2
	BY Wikinews	CC BY-SA	0.01
Encyclopedic	BY Wikipedia	CC BY-SA	37.0

Law Access Project (Caselaw Access Project), which covers over 6.5 million decisions published by state and federal courts throughout U.S. history.

- SW **Code:** We use the Github subset of the RedPajama dataset (Together, 2023), which contains code from Github repositories with three permissive software licenses: MIT, Apache, and BSD.
- SW BY **Conversation:** We source conversational text under permissive software licenses from the HackerNews (MIT license) and the Ubuntu IRC (Apache license) subsets of the Pile (Gao et al., 2020). We also use the Stackexchange subset of the RedPajama dataset (Together, 2023) and a Stackoverflow corpus from Kaggle,[4] both under the CC-BY-SA license.
- SW **Math:** We source mathematical text from the Deepmind Mathematics (Saxton et al., 2019) and the AMPS (Hendrycks et al., 2021) datasets, both of which are under the Apache license.
- PD BY **Science:** We source scientific text from ArXiv abstracts that are in the public domain (ArXiv, 2023). We also collect full-text articles from the Semantic Scholar Research Corpus (Lo et al., 2020, S2ORC), either licensed as public domain or CC-BY.
- PD **Books:** We source books from the Gutenberg corpus (Project Gutenberg), which are copyright-expired books that are in the public domain.
- PD BY **News:** We collect public domain news text from the English subset of the MOT corpus (Palen-Michel et al., 2022). We also collect text from Wikinews, which is under CC BY-SA.
- BY **Encyclopedic:** Finally, we include a large set of Wikipedia from the subset included in RedPajama (Together, 2023). We follow RedPajama in using Wikipedia snapshots from 20 languages even though the model primarily focuses on English.

Following Kandpal et al. (2022) and Lee et al. (2022), we deduplicate text using a document-level filter that considers n-gram overlap (Groeneveld, 2023). We first deduplicate within each domain to remove redundant documents from similar sources (e.g. Case Law and the Pile of Law), and then perform deduplication against the validation and test datasets of the Pile to avoid test leakage.

B.3 Data Analysis

To study the composition of OLC, we perform an n-gram based analysis of OLC domains against the validation data of the Pile. This enables us to better understand the dataset's domain shifts. We sample up to 10M tokens in each data source of OPEN LICENSE CORPUS and each subset of the Pile validation data. We then compute unigrams and bigrams across domains in both datasets (ignoring stopwords), and only consider n-grams that appear at least three times in each domain. For each validation domain of the Pile, we examine the *maximum* n-gram overlap across all OLC domains.

[4] https://www.kaggle.com/datasets/stackoverflow/stackoverflow

OLC domains have substantially less overlap with the validation data as compared to the Pile training domains: on average, the overlap between OLC domains and the validation domains is just 17%±9%, versus 28%±14% for the Pile training data. However, we find a large variance in overlap statistics across domains in OLC; we display the full matrix of n-gram overlap in Table 18. These results provide further evidence that models trained on OLC must handle larger domain shifts at test time than models trained on the Pile.

C Experimental Details

C.1 Details on the parametric component

LM architecture. We use 1.3B-parameter transformer LMs based on the LLaMA architecture (Touvron et al., 2023) as implemented in OpenLM.[5] Table 5 reports the hyperparameters for the parametric component of SILO. We keep these hyperparameters fixed for all parametric models that we report in this paper. We use the GPT-NeoX-20B tokenizer (Black et al., 2022), with 50432 BPE types.

Table 5: Basic hyperparameters for the parametric component of SILO.

Model	#L	#H	d_{model}	LR	Batch
1.3B	24	16	2048	1e-3	2.6M

Training. Each model is trained with 128 A100 GPUs across 16 nodes. We use 2,048 token sequences that are packed across document boundaries, and we pre-pend a beginning-of-text token to every document. We use weight decay of 0.1, the Adam optimizer with $\beta_2 = 0.95$, 2,000 steps of warmup, with a cosine learning rate scheduler. Following Muennighoff et al. (2023), we train for multiple epochs in each dataset, tracking validation perplexity every 10B tokens, and perform early stopping. As a result, we train our PD, PDSW and PDSWBY models for 60B, 250B, and 350B tokens in total, respectively.

Domain re-weighting. Since the distribution of OLC is highly skewed (§2), we perform a simple upweighting scheme where we upsample all data that accounts for less than 5% by a factor of 3×, which we found to work well after a sweep of different settings. More sophisticated domain weighting strategies (Xie et al., 2023) are of interest but beyond the scope of this work.

C.2 Details of adding the Nonparametric Component

C.2.1 Methodology

To incorporate the nonparametric component, we experiment with two widely-used retrieval methods (Figure 1): the k-nearest neighbors LM (kNN-LM; Khandelwal et al., 2020) and the retrieval-in-context approach (RIC-LM; Shi et al., 2023; Ram et al., 2023).

The ***k*-nearest neighbors language model (*k*NN-LM)** interpolates the next-token probability distribution from a parametric LM with a nonparametric distribution based on every token that is stored in a datastore. Given a text dataset consisting of N tokens $c_1 ... c_N$, a datastore is built by creating a key-value pair for every token c_i ($1 \leq i \leq N$); a value is c_i and a key k_i is $...c_{i-1}$, a prefix preceding c_i. At test time, given an input prefix x, the nonparametric distribution is computed by:

$$P_{k\text{NN}}(y \mid x) \propto \sum_{(k,v)\in\mathcal{D}} \mathbb{I}[v = y] \left(-d(\text{Enc}(k), \text{Enc}(x))\right).$$

Here, Enc is an encoder that maps a text into $\mathbb{R}^h$ and $d : \mathbb{R}^h \times \mathbb{R}^h \to \mathbb{R}$ is a distance function, where h is the hidden dimension. We follow Khandelwal et al. (2020) and use the output vector from the last layer of the transformers in the parametric LM as Enc, and L2 distance as d. The final model takes the kNN-LM output and interpolates it with the output from the parametric LM: $\lambda P_{\text{LM}}(y \mid x) + (1 - \lambda) P_{k\text{NN}}(y \mid x)$, where λ is a fixed hyperparameter between 0 and 1.

As an alternative to kNN-LM, the **retrieval-in-context language model (RIC-LM)** retrieves text blocks from a datastore and feeds them to the parametric LM in context. Given a dataset consisting of

[5] https://github.com/mlfoundations/open_lm

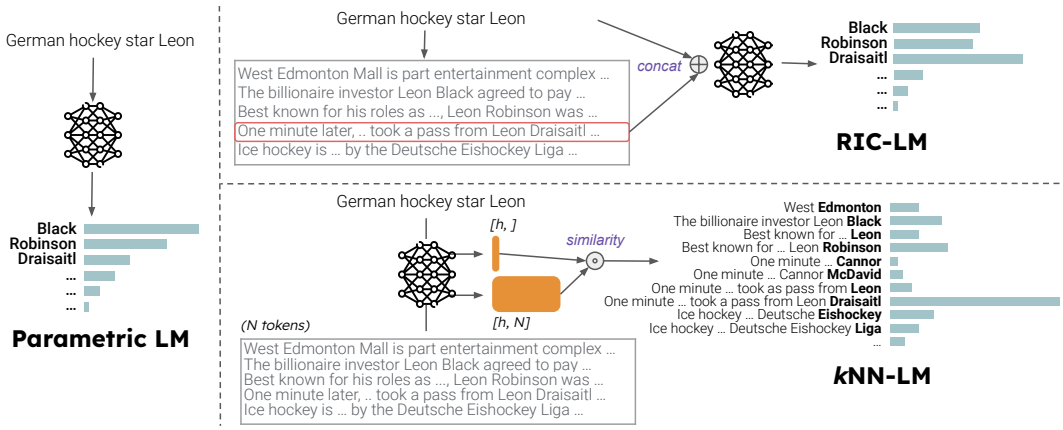

Figure 1: **An illustration of a parametric model and two retrieval methods we compare: RIC-LM and *k*NN-LM.** The orange boxes indicate representations of the input prefix and the tokens in the datastore, each in $\mathbb{R}^h$ and $\mathbb{R}^{h \times N}$, where h is a hidden dimension and N is the number of tokens in the datastore. The distribution from kNN-LM in the figure describes $P_{k\text{NN}}$; while omitted in the figure, the final output distribution from kNN-LM is an interpolation between $P_{k\text{NN}}$ and the distribution from the parametric LM. See §C.2 for more details of each method.

N tokens $c_1 ... c_N$, an index $\mathcal{D}$ is constructed by splitting the data into text blocks $b_1 ... b_M$, optionally with a sliding window. At test time, given an input prefix x, RIC-LM retrieves the most similar paragraph to the prefix $\hat{p} = \arg\max_{b \in \mathcal{D}} \text{sim}(b, x)$ and concatenates it before the prefix to produce $P_{\text{LM}}(y \mid \hat{b}, x)$. Here, sim is a function that computes a similarity score between two pieces of text; we use BM25 following Ram et al. (2023) who show that BM25 outperforms alternative dense retrieval.

Attribution and opt-out. Since elements in the datastore that contribute to the model prediction are transparent, both kNN-LM and RIC-LM offer inherent attributions. Moreover, data removed from the datastore is guaranteed not to contribute to any model predictions, allowing data owners to remove their data at the level of individual examples. Both are unique characteristics of nonparametric language models. While prior work studies post-hoc attribution to the data used for training model parameters (Koh & Liang, 2017; Han et al., 2023) and removing the effect of specific training examples from parameteric models (Cao & Yang, 2015; Jang et al., 2023b), they lack guarantees and are difficult to scale.

C.2.2 Implementation Details

For **kNN-LM**, each datastore consists of up to 1 billion h-dimensional vectors (h =2,048). We build an index for fast nearest neighbor search using FAISS (Johnson et al., 2019). We use IndexIVFPQ which quantizes vectors into 64-bytes and clusters them into 4,096 centroids, learned from 1 million sampled vectors, following Khandelwal et al. (2020). Instead of recomputing the exact L2 distance using the original embeddings, we use the L2 distance beteen quantized vectors returned by the FAISS index (ablations in Appendix D.3). Since their scale is not preserved, we use $\frac{d(\mathbf{x}_\text{q}, \mathbf{y}_\text{q})}{\tau}$ as a proxy of $d(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}_\text{q}$ and $\mathbf{y}_\text{q}$ are vectors quantized from $\mathbf{x}$ and $\mathbf{y}$. Hyperparameters, including k, λ, and τ, are chosen based on the validation data in a domain-specific manner.

For **RIC-LM**, each datastore consists of text blocks with a length of 1,024 and a sliding window of 512. We use BM25 from Pyserini (Lin et al., 2021).

Table 6 reports the datastore statistics for both RIC-LM and kNN-LM, as well as hyperparameter values for kNN-LM (λ, k, τ). Due to the resource constraints, the datastore size is capped to up to 10% of the PILE training data (and to 1024.0M tokens in the case of kNN-LM), but future work can investigate further scaling the datastore.

Table 6: Datastore statistics as well as hyperparameter values for kNN-LM. Underline indicates exact nearest neighbor search (instead of approximate) was performed for kNN-LM because the datastore is small enough. Hyperparameters are chosen based on the validation data of each domain.

Data	RIC-LM		kNN-LM			
	# tokens	# blocks	# tokens	λ	k	τ
Github	3084.3M	6.0M	1024.0M	0.2	128	10.0
NIH ExPorter	72.2M	0.1M	72.2M	0.3	32,768	20.0
Wikipedia	1177.9M	2.3M	1024.0M	0.3	4,096	20.0
CC News	382.2M	0.7M	382.2M	0.7	4,096	20.0
Books3	1424.7M	2.8M	1024.0M	0.2	4,096	25.0
Enron Emails	45.0M	0.1M	45.0M	0.5	4,096	1.0
Amazon	1214.3M	2.4M	1024.0M	0.5	32,768	20.0
MIMIC-III	519.5M	1.0M	519.5M	0.7	1,024	15.0

Table 7: Datastore and hyperparameter values for kNN-LM evaluated on downstream tasks. Hyperparameters (the choice of datastore, λ, k and τ) are chosen based on the validation data of each task.

Task	Datastore	λ	k	τ	Template	Label Words
AGN	Wikipedia	0.8	1024	3	The topic of the text is	politics, sports, business, technology
Dbpedia	Amazon	0.9	4096	3	The topic of the text is	company, school, artist, athlete ... (14 in total)
SST-2	IMDB	0.9	8192	3	It was	great, terrible
MR	IMDB	0.9	8192	3	It was	great, terrible
RT	IMDB	0.9	8192	3	It was	great, terrible
CR	Amazon	0.2	1024	3	It was	great, terrible
Yelp	IMDB	0.9	8192	3	It was	great, terrible
Amz	IMDB	0.9	8192	3	It was	great, terrible
RTE	Amazon	0.9	1024	3	true or false? answer:	true, false
HYP	CC News	0.1	1024	1	neutral or partisan? answer:?	neutral, partisan

C.3 Details of Language Modeling Perplexity Evaluation

We benchmark our models using language modeling perplexity on 14 domains that represent both in-domain and out-of-domain data with respect to different levels of OLC. This includes: public-domain legal documents from the **FreeLaw** Project subset of the the Pile (Gao et al., 2020), a held-out collection of books from the **Gutenberg** collection (Project Gutenberg), conversational text from the **Hacker News** subset of the Pile, held-out code files from the **Github** subset of the Pile (most of which are non-permissive licensed), scientific text of NIH Grant abstracts that are taken from the **NIH ExPorter** subset of the PILE, philosophy papers taken from the **PhilPapers** of the PILE, held-out English **Wikipedia** articles from the PILE, news articles from **CC-News** (Mackenzie et al., 2020), books from **BookCorpus2** which is an expanded version of Zhu et al. (2015), books from **Books3** by Presser (2020), random web-crawled pages from **OpenWebText2** (Gokaslan & Cohen, 2019; Gao et al., 2020), emails from the **Enron Emails** corpus (Klimt & Yang, 2004), **Amazon** product reviews from He & McAuley (2016), and finally clinical notes from **MIMIC-III** (Johnson et al., 2016) with personal identifiable information (PII) masked out. Our choice of domains highlights legal risks discussed in the earlier sections, e.g., CC-News, BookCorpus2, Books3 and Amazon reviews are mostly copyrighted, Github is mostly not permissively licensed,[6] and Enron Emails and MIMIC-III include private text.

We merge all text into one stream of text and split them into batches with a maximum sequence length of 1,024 and a sliding window of 512, a setup that is standard in prior language modeling literature (Baevski & Auli, 2019; Khandelwal et al., 2020). For MIMIC-III, which includes masked personally-identifiable information (PII), we filter out notes where more than 50% of tokens correspond to PII, and then exclude tokens corresponding to PII when computing perplexity.

[6] Kocetkov et al. (2023) estimates about 13% of the Github data is under MIT, Apache, and BSD, permissive licenses defined by Together (2023).

Table 8: Perplexity on the parametric LMs trained on PD, PDSW, and PDSWBY, as well as Pythia 1.4B, a model trained with similar amounts of compute but on non-permissive data. We use ■, ■, and ■ to indicate text that is in-domain, out-of-domain, or out-of-domain but has relevant data in-domain data (e.g., non-permissive Github code versus our permissive training code). Reported on the validation data; see Table 2 for results on the test data.

Eval data	PD	PDSW	PDSWBY	Pythia
FreeLaw	5.3	5.7	6.5	5.6
Gutenberg	14.6	11.9	13.4	12.7
HackerNews	36.6	12.1	13.2	12.5
Github	13.3	2.6	2.7	2.4
NIH ExPorter	28.6	19.3	15.1	11.2
PhilPapers	55.2	24.2	16.5	14.3
Wikipedia	27.9	19.7	11.1	9.0
CC News	30.8	21.3	19.3	10.9
BookCorpus2	25.2	19.2	20.2	12.8
Books3	25.9	18.7	18.1	12.4
OpenWebText2	38.1	21.2	18.8	11.5
Enron Emails	19.9	14.3	14.5	7.6
Amazon	81.9	34.7	37.0	22.8
MIMIC-III	18.2	16.4	13.6	11.5
Average	30.1	17.2	15.7	11.2

C.4 Details of Downstream Task Evaluation

We perform zero-shot prompting for nine text classification datasets: AGNews (Zhang et al., 2015), Yahoo (Zhang et al., 2015), Subj (Pang & Lee, 2004), SST-2 (Socher et al., 2013), MR (Pang & Lee, 2004), Rotten Tomatoes (RT), CR (Hu & Liu, 2004), Amazon polarity (Amz, McAuley & Leskovec (2013)) and RTE (Dagan et al., 2005). The tasks range from topic classification and sentiment analysis to subjectivity classification and textual entailment.

We use the templates and label words to map the task into a sentence completion problem, following the standard from literature (Radford et al., 2019; Brown et al., 2020; Holtzman et al., 2021). These templates and label words are taken from Shi et al. (2022), reported in Table 7. For both parametric LMs and kNN-LM, we apply the domain-conditional PMI scoring (Holtzman et al., 2021) for determining the probability of each label. For kNN-LM, we follow a method from Shi et al. (2022) which employs the fuzzy verbalizers to expand the token set associated with each output label in our task. Also following Shi et al. (2022), we use IMDB (Maas et al., 2011) with 8 million tokens as an additional datastore. We perform hyperparameter search on the validation dataset of each task, considering $k \in \{128, 512, 4196, 8192\}$, $\tau \in \{1, 3, 5, 10, 40, 80\}$, and different choice of datastores. The chosen hyperparameters are reported in Table 7.

D Additional Experimental Results

Table 8 reports perplexity of the parametric LMs on the validation data that is analogous to Table 2. Table 9 reports perplexity of both parametric and nonparametric LMs on the validation data that is analogous to Table 3. Findings based on the validation data and on the test data are largely consistent.

D.1 Experiments on Downstream Tasks

In order to verify if our findings on language modeling perplexity transfer to downstream tasks, we evaluate zero-shot performance of SILO and Pythia on ten text classification datasets whose domains are not covered by OLC. All models use PMI (Holtzman et al., 2021) for a better calibration of model outputs, and we use kNN-Prompt (Shi et al., 2022) for applying kNN-LM for downstream tasks. See §C.4 for the details.

Table 10 demonstrates that our earlier findings hold on all ten datasets: the parametric-only SILO largely underperforms Pythia; however, adding a datastore greatly improves performance, allowing performance of SILO to match that of Pythia.

Table 9: Perplexity of parametric LMs (Prm-only), kNN-LM and RIC-LM; ■ indicates in-domain; ■ indicates out-of-domain; ■ indicates out-of-domain but has relevant data in-domain. Reported on the validaiton data; see Table 3 for results on the test data.

Eval data	PDSW			Pythia
	Prm-only	kNN-LM	RIC-LM	Prm-only
Github	2.6	2.4	2.4	2.4
NIH ExPorter	19.3	14.9	18.5	11.2
Wikipedia	19.7	14.1	18.9	9.0
CC News	21.3	7.1	14.8	10.9
Books3	18.8	17.3	18.5	12.5
Enron Emails	14.3	6.7	11.1	7.6
Amazon	34.7	26.2	33.7	22.8
MIMIC-III	16.3	7.2	14.1	11.5
Average	18.4	12.0	16.5	11.0

Table 10: **Zero-shot performance on ten classification datasets.** kNN-LM allows performance of SILO to match that of Pythia by using a datastore. ■ indicates in-domain; ■ indicates out-of-domain.

Eval	SILO (PDSW)		Pythia
	Prm-only	kNN-LM	Prm-only
AGN	63.3	79.5	71.2
Dbpedia	36.9	41.3	39.1
SST-2	57.1	74.4	79.7
MR	55.1	79.0	79.9
RT	55.3	67.9	80.4
CR	64.6	83.1	80.3
Yelp	62.8	84.5	84.7
Amz	60.4	82.9	80.3
RTE	56.0	55.2	53.7
HYP	58.5	63.1	58.5
Avg	57.0	71.1	70.8

D.2 Ablations: Parametric Component

Effect of upsampling low-resource data. Since OPEN LICENSE CORPUS has an extremely skewed distribution of domains, we upsample less-representative domains during training. Table 11 (left) compares the models trained on PDSW with and without domain upweighting. In-domain datasets that are not upweighted, e.g., FreeLaw, see slight degration in performance. On out-of-doain datasets, there is no significant differences, although the model with upsampling is marginally better (19.6 vs. 19.7 when averaged over 9 out-of-domain datasets). We note that we did not tune the upweighting ratio nor explore alternative upweighting approaches (Xie et al., 2023) due to resource constraints, and leave them for future work.

59B tokens of source code significantly help. When using SW, a substantial 59.1% of the training data is actual source code. To determine SW provides such large gains, we also run an ablation where we include SW data but exclude all of the actual source code, i.e., we only include Hacker News, Ubuntu IRC, Deepmind Math, and AMPS on top of the PD data. This leaves models trained on 99.6B tokens for OLC (PDSW) and 40.7B for OLC (PDSW) excluding source code. Table 11 (right) report results on a subset of the validation domains. Including source code provide significant benefits for certain test datasets, e.g., nearly a 20 point improvement in perplexity on PhilPapers, likely because it significantly increases the size of the training data.

Performance gaps with Pythia are strongly correlated to domain shift. We hypothesize that the performance differences we see between our models and Pythia can be explained by domain shift between our corpus and the evaluation domains. To confirm these effects, we show a scatterplot that describes the relationship between ngram overlap of the OLC and the Pile (from §B.3 with

Table 11: **(Left)** Effect of re-weighting rare domains, comparing models trained on OLC (PDSW) with and without upsampling. **(Right)** Effect of SW data, with and without explicit source code—we train an LM with SW data but remove all of the actual source code (i.e., we leave Hacker News, Ubuntu IRC, Deepmind Math, and AMPS). Both tables report perplexity on the validation data.

Data	PDSW w/o upsampling	PDSW w upsampling
FreeLaw	4.9	5.7
Github	2.4	2.6
NIH ExPorter	20.0	19.3
PhilPapers	23.9	24.2
Wikipedia	19.9	19.7
CC News	21.8	21.3
BookCorpus2	19.4	19.2
OpenWebText2	21.0	21.2
Enron Emails	13.5	14.3
Amazon	35.7	34.7

Data	PD	PDSW w/o code	PDSW
FreeLaw	5.3	5.7	5.7
Github	13.3	8.2	2.6
NIH ExPorter	28.6	26.2	19.3
PhilPapers	55.2	36.4	24.2
Wikipedia	27.9	26.5	19.7
CC News	30.8	28.8	21.3
BookCorpus2	25.2	23.8	19.2
OpenWebText2	38.1	31.7	21.2
Enron Emails	19.9	18.5	14.3
Amazon	81.9	46.1	34.7

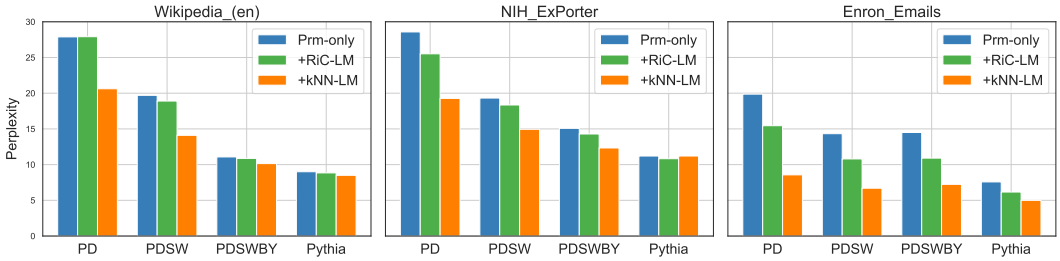

Figure 2: Results on different variants of models: PD, PDSW and PDSWBY variants of SILO as well as Pythia. Adding a nonparametric component through either RIC-LM and kNN-LM helps and kNN-LM is overall better than RIC-LM, consistently across all models and evaluation datasets.

the performance gap between PDSW and Pythia in Figure 6. There is a strong negative correlation between these two metric (r=-0.72, $p < 0.005$).

D.3 Ablations: Adding the Nonparametric Component

Impact of adding a nonparametric component across varying LMs. We compare parametric-only LM, RIC-LM and kNN-LM over four different LMs: PD, PDSW and PDSWBY variants of SILO as well as Pythia. Figure 2 reports their results on three evaluation datasets: Wikipedia, NIH ExPorter and Enron Emails. Findings from Section 3 hold across all models: both RIC-LM and kNN-LM are consistently better than the parametric-only LM, and kNN-LM overall achieves the best performance. For instance, kNN-LM allows the model to be comparable to or outperform the one-level relaxed variant, e.g., a PD-based kNN-LM is comparable to a PDSW-based parametric LM, and PDSW-based kNN-LM is comparable to a PDSWBY-based parametric LM.

Where does the remaining gap come from? Even when scaling the datastore with kNN-LM, SILO lags behind Pythia on a few domains. Moreover, a Pythia-based kNN-LM outperforms our model since kNN-LM improves Pythia as well. There are two possible points of failure in our model for these cases: either the parametric component (which outputs P_{LM}) struggles out-of-domain, or the encoder (that outputs $P_{k\text{NN}}$) struggles out-of-domain. To better understand which part of the model contributes to the gap we observe, we vary SILO with different choices for the parametric component and the encoder. We compare replacing either the parametric component or the encoder with Pythia. This setup allows us to measure the effects of the out-of-domain nature of our parametric component (which is only trained on PDSW subset of OLC) in each of these components.

Results in Figure 3 reveal that most performance gaps come from the LM: performance improves significantly when the parametric component is replaced with Pythia, given a fixed encoder. In contrast, performance improvement is relatively marginal when the encoder is replaced with Pythia, given a fixed parametric component. These results indicate that the parametric component, which

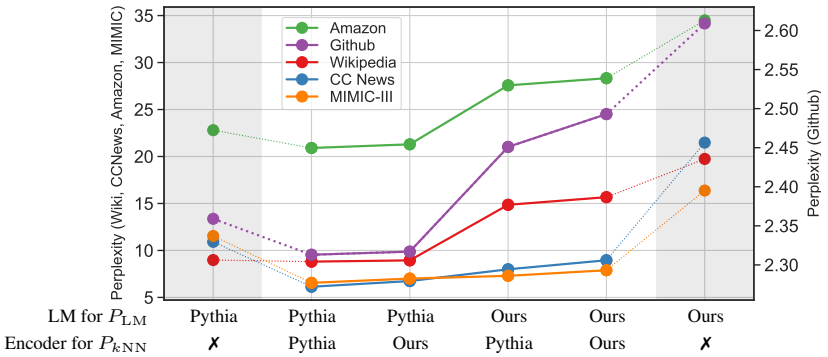

Figure 3: Impact of using different parameters on SILO. Perplexity on random 128K tokens from the validation data reported. The left-most and the right-most models are parametric models, and the other four models are kNN-LMs, using a datastore with 204.8 million tokens (20% of the datastore we use for the main experiments). *Ours* indicates our parametric model trained on the PDSW subset of OPEN LICENSE CORPUS. **Most of the performance degradation comes from using the out-of-domain parametric LM, rather than using the out-of-domain encoder.**

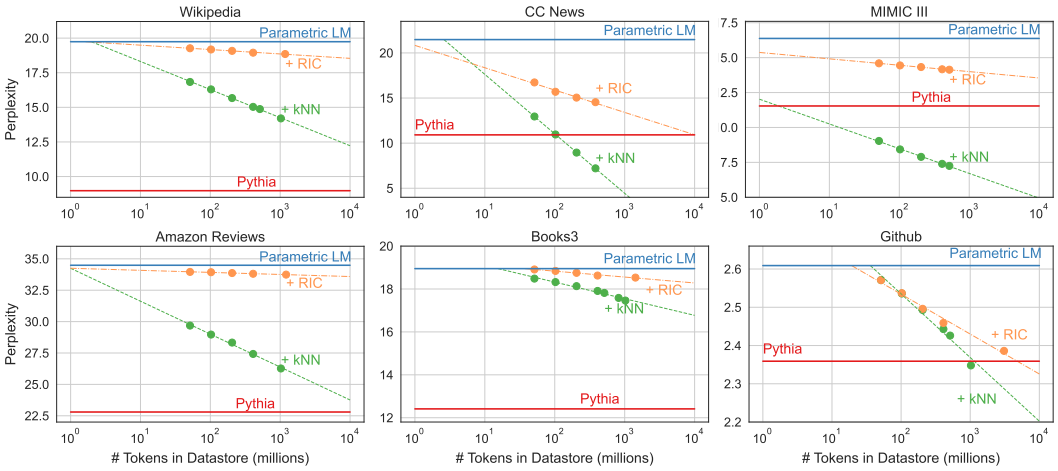

Figure 4: Impact of scaling the datastore of SILO (PDSW). Perplexity on random 128K tokens from the validation data reported. The rightmost dots for kNN-LM and RIC-LM in each figure correspond to the final models used in Table 3. **Scaling the test-time datastore consistently improves performance over all domains.**

gives P_{LM}, is quite sensitive to domain shift, but the encoder, which provides the nonparametric distribution $P_{k\text{NN}}$, is fairly robust to extreme domain shift. This also explains why kNN-LM generalizes better than RIC-LM, since RIC-LM is bottlenecked by the parametric component.

Effect of scaling the datastore in-domain and out-of-domain. Figure 4 shows the effect of scaling the datastore on domains that are out-of-distribution with respect to the training data. It shows that performance of both kNN-LM and RIC-LM rapidly improves as the datastore size grows, and kNN-LM improves more rapidly than RIC-LM does. See §3 for more detailed discussion.

Does this trend hold when the test domains are in-domain with respect to the training data? To answer this question, we examine effect of scaling the datastore with Pythia 1.4B, where all of our test datasets can be considered in-domain. Figure 5 reports the results: Pythia on the left, SILO (PDSW) on the right. Results show that both Pythia and SILO see consistent improvements from kNN-LM and RIC-LM as the datastore gets larger, although the slope is larger with SILO than with Pythia. kNN-LM scales better than RIC-LM does, resulting in kNN-LM outperforming RIC-LM with a

Table 12: Ablations on different variants of RIC-LMs. Perplexity on the validation data reported. *ensbl-10* is 10x slower than other three methods.

Data	PDSW				Pythia			
	basic	ensbl-10	concat-2	concat-next	basic	ensbl-10	concat-2	concat-next
CC News	14.8	13.5	17.0	18.8	8.2	7.9	9.2	9.9
Enron Emails	11.1	10.0	12.8	13.4	6.3	6.117	7.1	7.3

reasonably large datastore in most cases (with an exception of Pythia on Github, where RIC-LM outperforms kNN-LM with a reasonable size of a datastore).

Ablations on variants of RIC-LM. We explore four different variants of RIC-LM. (1) The **basic** is the method described in §C.2, which uses text blocks with a length of L each. At inference, it takes the top 1 text block from the datastore and feeds it to the LM, i.e., $P_{\text{LM}}(y|\hat{b}, x)$ where x is the input and $\hat{b}$ is the top 1 text block. (2) The **ensbl-*k*** ($k = 10$) variants is also based on text blocks with a length of L each. At inference, it takes the top k text blocks from the datastore, feeds each to the LM in parallel and aggregates the probability distributions, i.e., $\frac{1}{k}\sum_{1\leq i\leq k} P_{\text{LM}}(y|\hat{b}_i, x)$ where $\hat{b}_1 ... \hat{b}_k$ are the top k text blocks. This follows a method from Shi et al. (2023). (3) The **concat-*k*** ($k = 2$) variant uses text blocks with a length of $\frac{L}{k}$ each. At inference, it takes the top k text blocks from the datastore, concatenates them in a reverse order, and feeds it into the LM, e.g., $P_{\text{LM}}(y|\hat{b}_k, \cdots, \hat{b}_1, x)$ where $\hat{b}_1 ... \hat{b}_k$ are the top k text blocks. (4) The **concat-next** variant uses text blocks with a length of $\frac{L}{2}$ each. At inference, it takes the top 1 text block from the datastore, concatenates the text block and the subsequent text block in a datastore, and feeds it into the LM. This is based on the intuition that the continuation of the text block that is most similar to the query can be useful for the continuation of the query; Borgeaud et al. (2022) has explored a similar approach based on the same intuition. We use $L = 1024$ for all variants. The ensbl-k variant has run-time that is approximately k times of run-time of the basic, concat-k and concat-next.

Results are reported in Table 12. The concat-2 and concat-next variants perform poorly, while the ensbl-10 outperforms the basic variant. However, we reached the conclusion that the significant run-time cost (i.e., 20x compared to a parametric LM) does not justify the improvements, and thus, we primarily use the basic variant for the remaining experiments. Future work may involve re-evaluating models using the ensbl-k approach or enhancing its run-time efficiency.

Effect of different approximation methods for L2 distance. Prior work (Khandelwal et al., 2020) typically uses approximate nearest neighbor search to find the top k nearest neighbors, and then computes the exact L2 distance using the original vectors. However, this may be inefficient in disk memory usage and run-time speed, due to needing to store large, original vectors and access them on-disk. We thus explore a few alternatives: (1) quantizing the original vectors to compute the L2 distance (but less aggressively than quantization for the nearest neighbor search index, thus it provides different levels of approximations for search and for L2 distance), or (2) completely dropping the original vectors and relying on approximated L2 distance from the index with aggressive quantization. Table 13 shows that all approximation methods only marginally affect performance. For this reason, we use the most aggressive approximation that completely drops the original embeddings at the cost of about 0.5% lose in performance while using $< 2\%$ of the memory footprint. Future work may study more accurate and efficient approximation methods.

Table 13: Ablations on approximation methods on the validation data of Wikipedia, using the LM trained on PDSW and the datastore consisting of 51.2 million tokens (5% of the datastore in the main experiments). Relative disk memory usage reported.

Method	PPL	Disk use
Param-only	19.7	0.0
No approximation	16.4	1.0
Quantized (4x)	16.6	0.25
Quantized (8x)	16.6	0.125
Quantized (16x)	16.8	0.0625
IVFPQ approximation	16.8	0.0178

Runtime speed. Table 14 presents the runtime speed of the parametric LM, RIC-LM, and kNN-LM on the Wikipedia validation set. Speed is reported in tokens per second with a batch size of 1 using a single NVIDIA RTX 6000 GPU.

The results show that the parametric LM is notably faster than both RIC-LM and kNN-LM, and RIC-LM is faster than kNN-LM. Speed is slower as the datastore gets larger (for both RIC-LM and

kNN-LM) and the nearest neighbor search gets less accurate (for kNN-LM; indicated by the number of probe p). kNN-LM can eventually match RIC-LM's speed while surpassing its performance by using a smaller datastore and less accurate search, i.e., when using 102M tokens with $p = 1$.

However, the machine used for benchmarking has a very slow IO speed, leading to an underestimation of both RIC-LM and kNN-LM's runtime speed, and the comparison can significantly vary based on the hardware. Either way, kNN-LM is currently substantially slower than a parametric LM, leaving room for potential future improvements.

Table 14: Comparison in runtime speed (# tokens per second). p indicates the number of probe in FAISS ($p = 8$ in all experiments unless specified otherwise).

Method	PPL	# tokens/s
Param-only	19.7	1828.6
RIC-LM (51.2M)	19.3	812.7
RIC-LM (102.4M)	19.2	731.4
RIC-LM (204.8M)	19.1	588.5
RIC-LM (409.6M)	18.9	478.5
RIC-LM (1,178M)	18.9	419.7
kNN-LM (51.2M)	16.8	184.2
kNN-LM (102.4M)	16.3	112.0
kNN-LM (204.8M)	15.7	59.3
kNN-LM (409.6M)	15.0	31.8
kNN-LM (1,024M)	14.2	14.2
kNN-LM (102M, $p = 1$)	16.7	560.8
kNN-LM (1,024M, $p = 1$)	14.6	71.1
kNN-LM (1,024M, $p = 2$)	14.4	45.5
kNN-LM (1,024M, $p = 4$)	14.2	27.0

D.4 Examples of Data Attribution and Opt-Out

The design of SILO can better align with various data-use regulations by providing mechanisms for data attribution during inference and for data owners to remove their data from the model at any time.

Data opt-out. To showcase the impact of opt-out on model performance, we conduct experiments with J. K. Rowling's Harry Potter series. We first identify all seven Harry Potter books from the Books3 corpus of the Pile. For each book, we calculate the perplexity of SILO using two 1B-token datastores on Books3, but one including the remaining six Harry Potter books and the other excluding any Harry Potter books. This experiment aims to see whether excluding Harry Potter books from the former datastore can reduce the likelihood of generating the left-out book.

Table 15 shows the results. SILO with Harry Potter books in the datastore (third column) effectively improves perplexity for all seven books, closing the gap between the PDSW model (first column) and Pythia (fourth column). However, when the Harry Potter books are removed from the datastore (second column), the perplexity gets worse, approaching that of the parametric-only LM. This illustrates that eliminating the effect of the Harry Potter books from the model substantially reduces the likelihood of generating the leave-out book.

Table 15: **The effect of data opt-out**, reporting perplexity. *w/ HP* and *w/o HP* indicate that the datastore includes or excludes Harry Potter books, respectively. The number (1 to 7) indicates a different book from the Harry Potter series used as the eval data; this eval book is not included in the datastore.

Eval	SILO (PDSW)			Pythia
	Prm-only	kNN-LM w/o HP	kNN-LM w/ HP	Prm-only
1	15.9	15.2	13.0	9.6
2	17.7	16.7	12.4	10.0
3	16.5	15.6	11.4	9.5
4	17.7	16.8	12.9	10.1
5	17.8	16.9	13.2	10.2
6	17.4	16.5	12.8	10.1
7	18.8	17.8	15.1	10.9
Avg	17.4	16.5	12.9	10.1

Attribution. To show the attribution feature of our model, Table 16 provides qualitative examples on the top-1 context retrieved by SILO. The model is able to assign a high probability to the ground truth token by retrieving highly relevant context. It achieves this by leveraging the unique characteristics of the text within the datastore, such as recognizing that *Azkaban* refers to the prison and *green light* is associated with the *Killing Curse* in the Harry Potter books.

More qualitative examples on Github, news and emails are illustrated in Table 17. The model is able to assign a high probability to the ground truth token by retrieving highly relevant context, e.g., given the context (hockey) and the first name of the player, being able to retrieve the last name of the player, given the context (a show and its host), being able to complete the quote. These examples also highlight that a nonparametric approach addresses specific legal risks that we have discussed earlier, e.g., it assigns per-token attribution for free, and can provide a copyright notice when part of copyrighted text is being used for the probability distribution.

Test Prefix 'I - what - *dragons*?' spluttered the Prime Minister. 'Yes, three,' said Fudge. 'And a sphinx. Well, good day to you.' The Prime Minister hoped beyond hope that dragons and sphinxes would be the worst of it, but no. Less than two years later, Fudge had erupted out of the fire yet again, this time with the news that there had been a mass breakout from **Test Continuation** **Azkaban**. **Retrieved Prefix** 'D' you know Crouch, then?' said Harry. Sirius' face darkened. He suddenly looked as menacing as the night when Harry had first met him, the night when Harry had still believed Sirius to be a murderer. 'Oh, I know Crouch all right,' he said quietly. 'He was the one who gave me the order to be sent to **Retrieved Continuation** **Azkaban** - without a trial.'
Test Prefix Terror tore at Harry's heart... he had to get to Dumbledore and he had to catch Snape... somehow the two things were linked... he could reverse what had happened if he had them both together... Dumbledore could not have died... (...) Harry felt Greyback collapse against him; with a stupendous effort he pushed the werewolf off and onto the floor as a jet of **Test Continuation** **green** light **Retrieved Prefix** Voldemort was ready. As Harry shouted, "Expelliarmus!" Voldemort cried, "Avada Kedavra!" A jet of **Retrieved Continuation** **green** light issued from Voldemort's wand

Table 16: **Attribution examples on Harry Potter books.** We show the top-1 retrieved context of SILO (PDSW). **Red underlined text** indicates the next token that immediately follows the prefix. In both examples, the test data is from the sixth novel and the retrieved context is from the fourth novel in the Harry Potter series. In the series, *Azkaban* is the notorious wizarding prison, and the *green light* is a distinct characteristic of the Killing Curse, *Avada Kedavra*.

E Discussion & Future Work

Our work suggests that it is possible to improve the tradeoff between legal risk and model performance when training LMs. Our approach provides new options for model designers to mitigate the legal risk of LMs, and empowers stakeholders to have more control over the data that drives these systems. We point out a number of rich areas for future work, beyond what was mentioned throughout the paper:

Addressing the limitations of SILO. SILO does not completely eliminate legal risk. Instead, it provides users more control over the model's generated content and functionalities to better align with legal regulations. For instance, SILO does not remove the need for obtaining permission to use copyrighted content in a datastore when providing attribution is not sufficient, but its opt-out capabilities can strengthen fair use defense. Moreover, SILO does not prevent copying copyright content from a datastore, but it offers a way to prevent generating sensitive text (Huang et al., 2023) or prevent copying the content verbatim. These functionalities increase the likelihood of a successful fair use defense if used appropriately.

Furthermore, while SILO mitigates copyright and privacy risks, it may exacerbate certain fairness issues, like toxicity towards marginalized groups and racial biases, especially due to the prevalence of older copyright-expired books in the training data. Exploring the balance between legal risk mitigation and fairness is an important future direction.

Finally, our study relies on explicit metadata to identify licenses, which may lead to underestimates of the amount and diversity of permissively licensed text actually available on the web. Future research may investigate *inferring* data licenses from documents in web crawl at scale, which may be an effective way to build more heterogeneous, permissively licensed corpora.

Introducing novel data licensing approaches. SILO introduces the possibility for data owners to set different levels of permissivity for learning parameters and for including in a nonparametric datastore. A data owner might choose to be more permissive about including data in the datastore due to its ease of removal, ensuring that the excluded data has no influence on model predictions anymore, and its ability to provide per-prediction attribution. Moreover, we envision that SILO could provide a path forward for data owners to get properly credited (or be paid directly) every time their data in a datastore contributes to a prediction. This is orthogonal to recent work that circumvented copyright issues by licensing out training data from data creators (Yu et al., 2023).

Investigating other copyright risk mitigation strategies. It is critical to continue to develop new techniques that use copyrighted data while protecting the rights of data owners and subjects. In addition to nonparametric approaches, there are many other ways to achieve these goals. First, one could train LMs on copyrighted content but filter and guide their outputs towards text that is non-infringing (Henderson et al., 2023). Second, training models with differential privacy (Dwork et al., 2006; Abadi et al., 2016) or near-access freeness (Vyas et al., 2023) may prevent them from regenerating individual details of copyright data. Finally, one could provide attributions for standard base LMs using post-hoc attribution methods, e.g., influence functions (Koh & Liang, 2017), rather than switching the model class to a retrieval-based model. All of these methods are complementary and orthogonal to our proposed approach.

Generalizing SILO as a modular language model. Our work is closely related to recent studies on modular LMs, which have specialized parameters (or *experts*) trained on different domains (Gururangan et al., 2022; Li et al., 2022; Gururangan et al., 2023), languages (Pfeiffer et al., 2020, 2022), or tasks (Chen et al., 2022b; Jang et al., 2023a). Our work extends modular LMs to include nonparametric datastores, and focuses on *specializing* different parts of the model to low- and high-risk subsets of the training data. Legal risks may also be mitigated with a collection of parametric expert models that are specialized to low- and high-risk data. Future work may explore this possibility as well as the usefulness of combining a nonparametric datastore with parametric experts.

Extending SILO to other modalities. While this work focuses on text-only models, similar methods to ours could apply to other domains and modalities. For instance, it might be possible to build permissive text-to-image generative models (Rombach et al., 2022) using compartmentalized public domain pre-training and retrieval-augmentation (Chen et al., 2022a; Golatkar et al., 2023). We believe such approaches are especially promising because there are many sources of public domain data in other modalities, e.g., images, speech, video, and more.

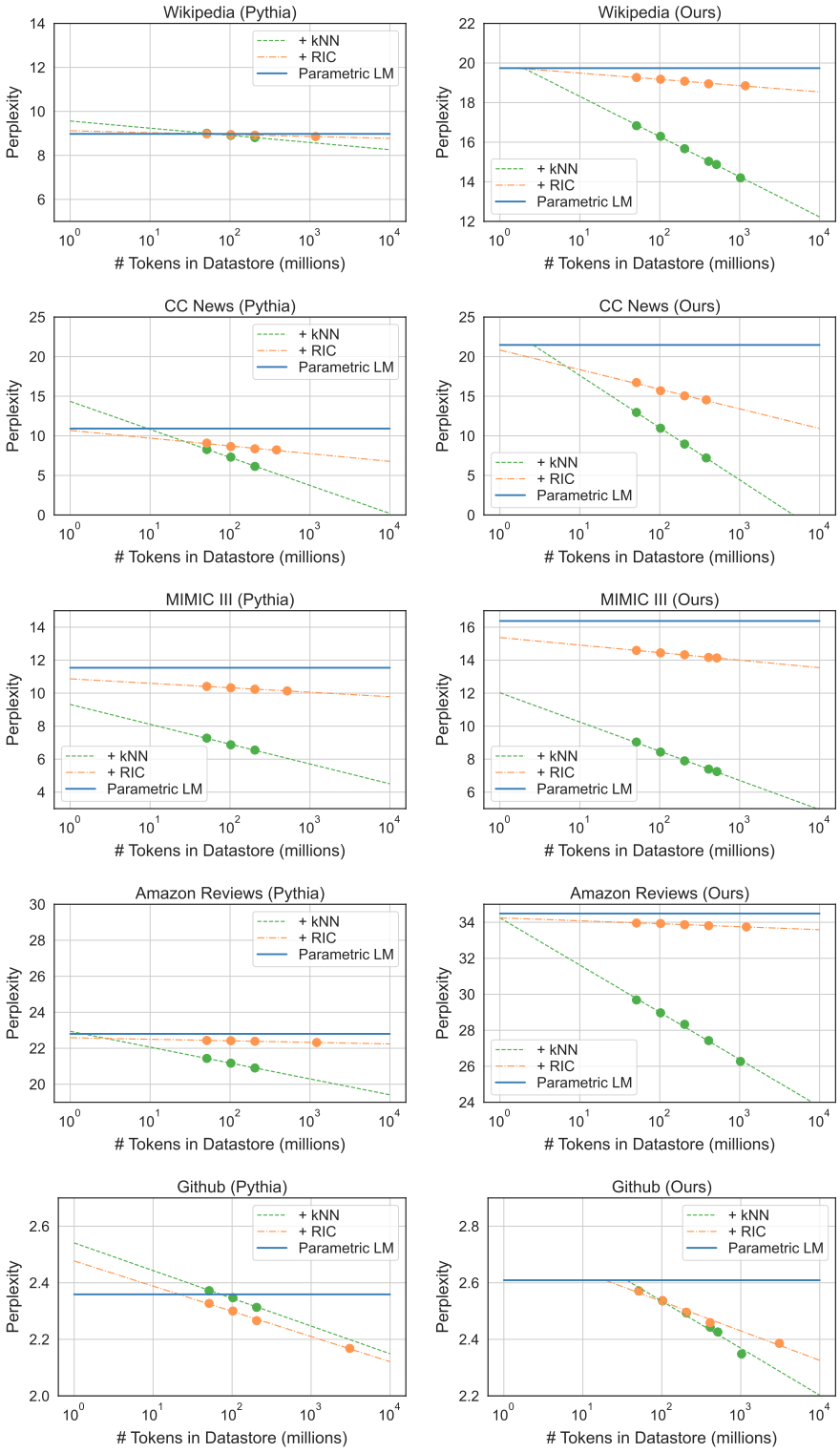

Figure 5: Comparison between parametric LM, RIC-LM and kNN-LM on five domains, with Pythia (left) and SILO PDSW (right), respectively. Perplexity on random 128K tokens from the validation data reported.

Table 17: Qualitative examples of retrieved context of our model. **Red underlined text** indicates the next token that immediately follows the prefix. The first two are from Github; the next two are from Enron Emails; and the last two are from CC News.

Test Prefix
include '../lib/admin.defines.php';
include '../lib/admin.module.access.php';
include '../lib/admin.smarty.php';
if (! has_rights (
Test Continuation **AC**X_BILLING)) { Header . . .
Retrieved Prefix
(...)
* You should have received a copy of the GNU Affero General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*
*
**/
if (! has_rights (
Retrieved Continuation **AC**X_ACCESS)) { Header ...

Test Prefix
0x5f #define S5K4AA_DEFAULT_BRIGHTNESS 0x10
/**********/
/* Kernel
Test Continuation **module** parameters */ extern int force_sensor; ...
Retrieved Prefix
* Copyright © 2011-2013 Jozsef Kadlecsik <kadlec@blackhold.kfki.hu>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License version 2 as
* published by the Free Software Foundation.
*/
/* Kernel
Retrieved Continuation **module** implementing an IP set type: . . .

Test Prefix . . . Mark or credit about hedge funds? Sara
Sara Shackleton
Enron North America Corp.
[Address]
[Phone number]
[Email address]
— Forwarded by Sara Shacleton/HOU/ECT on 01/2023/2022 05:41PM —
Tana
Test Continuation **Jones** 12/14/2000
Retrieved Prefix ... Food will be provided! Tana: Please feel free to extend the invitation to any Enron employees who may be interested in te presentation. 1st come, 1st serve. Thanks, Sylvia. — Forwarded by Sylvia Hu/Corp/Enron on 07/14/2000 03:17PM — Tana
Retrieved Continuation **Jones**@ECT. 07/13/2000

Test Prefix Ken Lay and Jeff Skilling were interviewed on CNNfn to discuss the succession of Jeff to CEO of Enron. (...) and then choose "Enron's Succession Plan.". The interview will be available every 15 minutes
Test Continuation **through** Friday, Dec. 15.
Retrieved Prefix Did you miss Jeff on CNBC "Street Signs" yesterday? Not to worry. (...) and then choose > "Skilling CNBC.". The interview will be available every ten minutes
Retrieved Continuation **through** > Wednesday, Dec. 6.

Test Prefix . . . The teams toured the city, explored west Edmonton mall and also got to take in an Oilers practice where they met German hockey star Leon
Test Continuation **Draisait**l
Retrieved Prefix One minute and 19 seconds later, Cannor McDavid took a pass from Leon
Retrieved Continuation **Draisait**l

Test Prefix ... Foley on RAW's run-time issues. Claiming that having the show run so late is one of the reasons why the final hour of RAW tends to struggle, Foley didn't end there. "No one else at 10:30pm is a
Test Continuation **PG** show. I won't say that across
Retrieved Prefix . . . way to the ring' podcast Foley cited RAW's duration and RG rating as hindrances to the show's popularity. Here's what he had to say: "Sometiems we try to look into the reasons why the third hour doesn't perform as well as the first two, and I'm like 'well that's because people go to bed! No one else at 10:30pm is a
Retrieved Continuation **PG** show. I won't say that across

Table 18: Unigram and bigram overlap between the domain of the Pile validation data and the domains of OPEN LICENSE CORPUS.

Dataset	BookCorpus2	Books3	Enron Emails	FreeLaw	Github	Gutenberg (PG-19)
ccby_law	0.02	0.04	0.02	0.08	0.02	0.03
ccby_s2orc	0.05	0.10	0.03	0.06	0.05	0.07
ccby_stackexchange	0.05	0.07	0.03	0.04	0.16	0.05
ccby_stackoverflow	0.03	0.05	0.01	0.03	0.07	0.03
ccby_wikinews	0.07	0.15	0.02	0.08	0.03	0.09
ccby_wikipedia	0.05	0.11	0.02	0.06	0.03	0.08
pd_books	0.13	0.26	0.03	0.07	0.04	0.33
pd_law	0.05	0.10	0.02	0.35	0.03	0.07
pd_news	0.06	0.14	0.02	0.07	0.02	0.08
pd_s2orc	0.08	0.15	0.04	0.08	0.05	0.13
sw_amps_math	0.01	0.02	0.01	0.01	0.02	0.01
sw_dm_math	0.00	0.01	0.00	0.01	0.01	0.01
sw_github	0.04	0.04	0.03	0.03	0.24	0.04
sw_hackernews	0.09	0.18	0.03	0.06	0.06	0.10
sw_ubuntu_irc	0.12	0.11	0.06	0.04	0.08	0.09
Dataset	**OpenWebText2**	**PhilPapers**	**Wikipedia (en)**	**cc-news**	**new-amazon**	**HackerNews**
ccby_law	0.02	0.03	0.05	0.05	0.02	0.03
ccby_s2orc	0.05	0.11	0.02	0.06	0.06	0.07
ccby_stackexchange	0.06	0.08	0.04	0.05	0.06	0.10
ccby_stackoverflow	0.05	0.07	0.02	0.03	0.07	0.06
ccby_wikinews	0.09	0.22	0.02	0.04	0.09	0.08
ccby_wikipedia	0.06	0.17	0.02	0.07	0.07	0.06
pd_books	0.16	0.15	0.03	0.11	0.11	0.09
pd_law	0.06	0.10	0.02	0.06	0.06	0.05
pd_news	0.08	0.21	0.02	0.06	0.09	0.06
pd_s2orc	0.08	0.13	0.03	0.09	0.08	0.09
sw_amps_math	0.01	0.02	0.01	0.01	0.01	0.02
sw_dm_math	0.00	0.01	0.00	0.00	0.00	0.00
sw_github	0.04	0.04	0.03	0.05	0.03	0.06
sw_hackernews	0.14	0.20	0.04	0.06	0.16	0.19
sw_ubuntu_irc	0.13	0.10	0.10	0.10	0.08	0.18

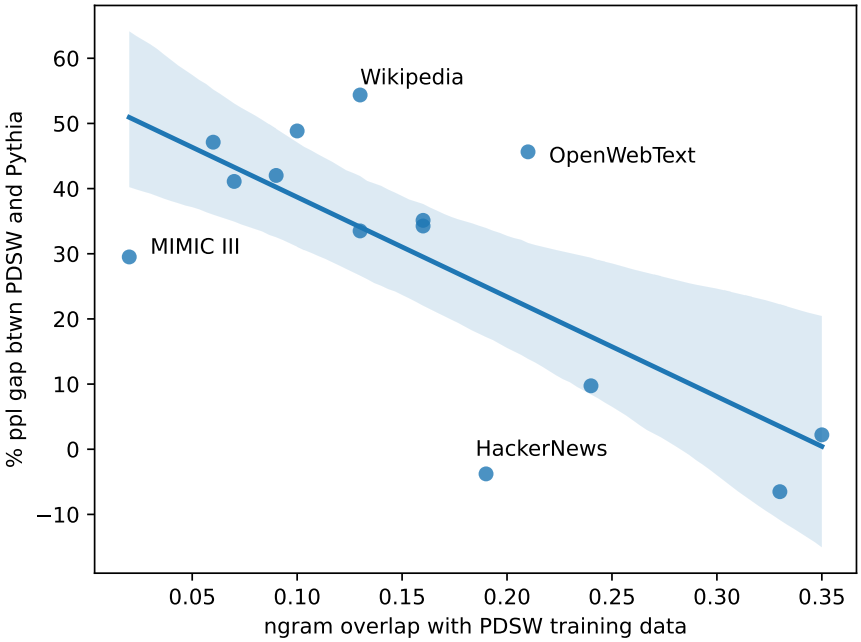

Figure 6: There is a strong negative correlation between ngram overlap of a domain with the PDSW training data and the perplexity gap between the PDSW LM and Pythia (r=-0.72, $p < 0.005$).