SEMDEDUP: DATA-EFFICIENT LEARNING AT WEB-SCALE THROUGH SEMANTIC DEDUPLICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Progress in machine learning has been driven in large part by massive increases in data. However, large web-scale datasets such as LAION (Schuhmann et al., 2022b) are largely uncurated beyond searches for exact duplicates, potentially leaving much redundancy. Here, we introduce SemDeDup, a method which leverages embeddings from pre-trained models to identify and remove "semantic duplicates": data pairs which are semantically similar, but not exactly identical. Removing semantic duplicates preserves performance and speeds up learning. Analyzing a subset of LAION, we show that SemDeDup can remove 50% of the data with minimal performance loss, effectively halving training time. Moreover performance increases out of distribution. Also, analyzing language models trained on C4, a partially curated dataset, we show that SemDeDup improves over prior approaches. SemDeDup provides an example of how simple ways of leveraging quality embeddings can be used to make models learn faster with less data.

1 INTRODUCTION

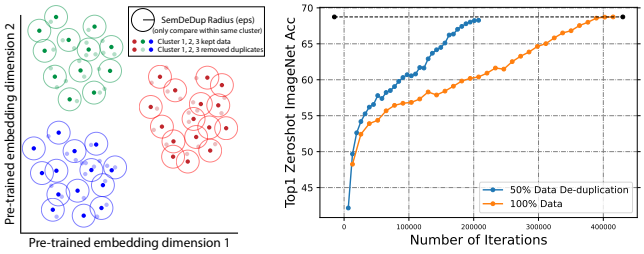


Figure 1: **Data efficiency from semantic deduplication (SemDeDup)** Left: A schematic of the SemDeDup algorithm which efficiently removes semantic duplicates from web-scale data. Right: When SemDeDup removes 50% of the LAION-440M dataset, training on this semantically *nonredundant* subset achieves almost the *same* performance as training on the *entire* 440M dataset. Also, training speed is *twice* as fast and completes in *half* the time.

A primary driver of recent success in machine learning has been the rise of self-supervised learning (SSL) scaled to ever larger models and unlabelled datasets (Hestness et al., 2017; Kaplan et al., 2020; Henighan et al., 2020; Rosenfeld et al., 2020; Gordon et al., 2021; Hernandez et al., 2021; Zhai et al., 2021; Hoffmann et al., 2022). In particular, modern large datasets are often derived at global web-scale and are generally unfiltered, with the exception of NSFW filters. One such public dataset is LAION (Schuhmann et al., 2022b), a multi-modal dataset of 5 billion image/text pairs. Multi-modal models such as CLIP (Radford et al., 2021b) are trained for many epochs on these large datasets achieving impressive performance, but at the cost of extremely long training durations.

The critical role of large datasets has led to increasing interest in scaling laws which enable us to predict how model performance changes given more data and/or parameters, leading to the observation that test error generally scales as a power law with respect to data quantity (Kaplan et al., 2020). Power law scaling, however, is unsustainable as diminishing marginal returns are quickly hit such that ever increasing amounts of data are required to achieve ever diminishing improvements in

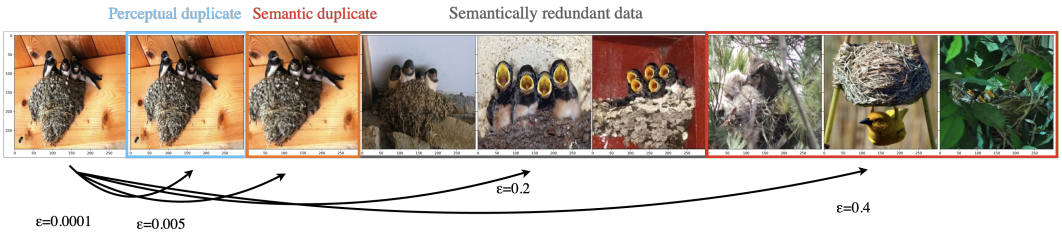


Figure 2: **Mapping cosine similarity to perceptual and semantic similarity.** We visualize pairs of images with cosine similarity $1 - \epsilon$ in the CLIP image encoder embedding space. The left most image is a random seed image from LAION, while the remaining images are sorted by their dissimilarity ϵ to the seed image. As ϵ increases from left to right, we move from perceptual to semantic duplicates, while at large values of ϵ we see semantically redundant pairs. Note the red labelled "semantic duplicate" is a view of the original left-most seed image from a slightly different perspective.

performance. Notably, many of these models appear never to converge, as test performance continues to increase even after 10s of passes through these massive datasets (Ilharco et al., 2021; Aghajanyan et al., 2023). This result suggests that our best models are underfitting, likely as a result of spending an increasing fraction of learning time focusing on redundant data.

Improving data efficiency is therefore quite impactful, either by enabling models to achieve the same performance much faster or to achieve better performance given the same computational budget. This has inspired recent work which suggests that by pruning data according to an intelligent criterion, power law scaling with respect to data can be beat and, given an optimal data ranking metric, exponential scaling might in principle be achieved (Sorscher et al., 2022). Recent explorations of this direction have shown promising results, with some works able to reduce data size by almost 5-fold with minimal performance loss (Radenovic et al., 2023).

Optimal approaches to select data might focus on one of several different classes of examples to be removed, roughly ordered by the complexity of their discovery:

1. **Perceptual duplicates**: We loosely define such data pairs to be perceptually identical to a typical human observer. These may be exact duplicates or inputs with slight, human impreceptible distortions.
2. **Semantic duplicates**: These are examples which contain largely identical information content, but remain perceptually distinct. For example, a pair of image views which are derived from the same image, but feature different margins, aspect ratios, or color distributions could be considered semantic duplicates. Such pairs would rarely, if ever, be detected by exact duplicate filters as they are far apart in pixel/token space.
3. **Semantically redundant data**: in contrast to semantic duplicates, semantically redundant data are not derived from the same underlying objects and would be clearly distinguishable to a human. However, the information contained in such examples may still contain substantial overlap. For example, consider the case of two different images of two different golden retrievers in two different parks. These images are neither perceptually nor semantically identical as the content of the images differ. Each additional semantically redundant data point will provide less and less new information, eventually converging to near-zero information gained from additional such data.
4. **Misleading data**: these are data which rather than providing zero information (as in the previous categories) provide negative or *harmful* signal, such that removing these data actually improves performance rather than having a neutral effect. While such data are easy to conceive of in supervised learning (i.e. mislabeled examples), it is much less clear what such examples may be in the context of self-supervised learning.

In this work, we focus on the category of semantic duplicates: data which are semantically highly similar but which would be difficult to discover using simple deduplication approaches. These data points are challenging to identify because distance measures in input space are unlikely to uncover semantic duplicates. To overcome this limitation, we leverage pre-trained foundation models to

compare data similarity in the learned embedding space rather than in input space. Comparing every data point to every other data point, however, is intractable, especially for web-scale datasets containing billions of examples. To make this computation possible, we use the clustering approach described in Sorscher et al. (2022) to segment the embedding space, allowing us to only search for duplicate pairs within a cluster. Using this approach, we make the following contributions:

- We propose SemDeDup (Fig. 1, left), a simple, yet effective and computationally tractable way to identify semantic duplicates. Using this approach, we show that large web-scale datasets such as LAION contain large numbers of semantic duplicates, with 50% of examples containing at least one semantic duplicate.
- Large fractions of semantic duplicates can be removed with little-to-no performance impact, greatly increasing training efficiency. We reduced the size of our LAION training set by 50% with minimal performance loss, achieving nearly the same performance 2x faster (Fig. 1, right), and moreover *improved* performance out-of-distribution.
- We apply SemDeDup to C4, beating prior SoTA deduplication, providing efficiency gains 10-15% for language model training, and sometimes even improving performance.

2 SEMDEDUP: DEFINING AND IDENTIFYING SEMANTIC DUPLICATES

While identifying perceptual duplicates can be easily done in input space, identifying semantic duplicates is more difficult as they maybe distant in either pixel or token space. To identify these pairs, we leverage the embedding space of a large pre-trained foundation model to provide a more semantically meaningful distance metric (Fig. 1, left). First, we embed each data point using a foundation model (OpenCLIP (Ilharco et al., 2021) for images and OPT (Zhang et al., 2022) for language. We then cluster the embeddings into $\mathcal{K}$ clusters via k-means ($\mathcal{K} = 50,000$). Within each cluster, we compute all pairwise cosine similarities and set a threshold cosine similarity (1-ϵ) above which data pairs are considered semantic duplicates. Finally, from each group of semantic duplicates within a cluster, we keep the image with the lowest cosine similarity to the cluster centroid and remove the rest. For CLIP experiments, we note that our method only considers images and ignores the captions.

3 SEMDEDUP ON LAION

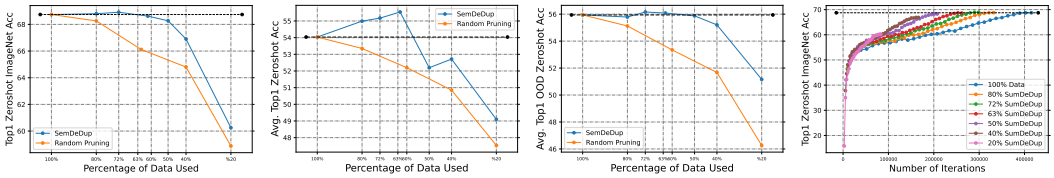
We first show that the LAION dataset contains extreme amounts of semantic redundancy (Section 3.1) and provide examples of the semantic duplicates discovered by SemDeDup (Section D.1). Most critically, we demonstrate that removing the semantic duplicates discovered by SemDeDup has minimal to no impact on converged performance and increases learning speed (Section 3.3). In Appendix B we show that SemDeDup improves performance of language models over prior methods.

3.1 EXTREME SEMANTIC REDUNDANCY AT WEB-SCALE

How many semantically redundant pairs are there in LAION? Remarkably, we find that even tiny thresholds ϵ lead SemDeDup to remove large fractions of data in LAION440M (Fig. A4a), showing that LAION-440M contains large quantities of semantic duplicates. Surprisingly, 30% of images in LAION-440M have a semantic duplicate at the highly stringent distance threshold of $\epsilon = 0.00095$, while 50% have a duplicate at the tight threshold of $\epsilon = 0.03$ (Fig. A4c). Moreover, a histogram of pairwise cosine similarity in LAION-440M (Fig. A4d) reveals a high density of pairs at high cosine similarity, including a large contribution at 1, reflecting highly similar semantic duplicates. These results demonstrate that LAION-440M contains large amounts of semantic redundancy.

3.2 WHAT DO SEMANTIC DUPLICATES LOOK LIKE?

In Fig. 2, we show examples of semantic duplicates found at different thresholds ϵ. At extremely low values of ϵ we find perceptual duplicates, and at slightly higher values of ϵ, we find semantic duplicates, which are the same image but with distortions which evade exact de-duplication approaches such as different margins, crops, aspect ratios, and color filters, or slighly different peripheral details. Fig.



(a) (b) (c) (d)

Figure 3: **SemDeDup allows better average zero-shot accuracy across 31 tasks with less data and faster pre-training. (a)**: Down to using only 50% of LAION-440M for pre-training CLIP, we are able match the zero-shot ImageNet accuracy of the baseline model trained on 100% of the data (black dashed line) with a small drop of 0.47% only, while we outperform the baseline model with only 63% of data. **(b)**: Average zero-shot performance for CLIP measured on 25 datasets *improves* down to 63% of the pre-training data, yielding *better* performance with almost $1.6\times$ *faster* pre-training. **(c):** Zeroshot validation accuracy averaged over 6 ImageNet-1k OOD datasets. We outperform the baseline model on OOD tasks with only 63% of pre-training. **(d):** We track zeroshot ImageNet-1K performance as a function of LAION-440M pre-training iterations at different deduplication thresholds. With SemDeDup, models converge in far less iterations.

A11, A12, and A13 show examples of clusters that are semantically deduplicated at increasing levels of ϵ, clearly indicating more semantic diversity in deduplicated clusters as ϵ increases.

3.3 Training on semantically deduplicated data improves efficiency

If SemDeDup is effective at finding semantic duplicates, we should be able to remove these duplicates with minimal performance impact. To test this, we train CLIP models on subsets of LAION-440M deduplicated at different thresholds ϵ. As ϵ rises, we retain a smaller fraction of data. In Fig. 3 (a), we plot the top-1 zero-shot accuracy of our CLIP models on ImageNet-1k. Encouragingly, we found that SemDeDup can remove up to 37% of LAION-440M with no performance drop, and 50% with minimal performance drop ($< 0.5\%$). In contrast, randomly removing data results in much larger drops. In Fig. 3 (b), we show the average zero-shot performance across 25 tasks, finding that on average, performance increased on de-duplicated data. See Table A1 for detailed performance on all 25 tasks at 6 deduplication thresholds as well as 1 baseline and 4 random controls. See also Fig. A7 for performance on 25 individual tasks.

We also evaluated out-of-distribution robustness on 6 datasets commonly used for this task: ImageNet-A, ImageNet-O (Hendrycks et al., 2021b), Imagenet-R (Hendrycks et al., 2021a), Imagenet-sketch (Wang et al., 2019), ImageNetV2 (Recht et al., 2019), and ObjectNet (Barbu et al., 2019). We again found that SemDeDup *increased* performance when *removing* 37% of the data, and matched performance when 50% was removed as shown in Fig. 3 (c). See Table A2 for detailed performance on 6 OOD tasks at 6 deduplication thresholds as well as 1 baseline and 4 random controls. See also Fig. A8 for performance on the 6 individual tasks. We emphasize that SemDeDup achieves these results on LAION-440M, an already highly curated dataset derived from LAION-2B which was found to have similar performance despite the almost five-fold reduction in data (Radenovic et al., 2023). However, to ensure that this curated subset did not bias our results, we also evaluated on LAION-233M, an uncurated subset of LAION-2B, finding qualitatively similar results (Fig. A5).

Because SemDeDup reduces the number of training points, it enables substantially faster training. In Fig. 3 (d), we plot the top-1 zero-shot accuracy on ImageNet-1k as a function of the number of iterations for different deduplication thresholds ϵ. Notably, models trained on deduplicated data reach convergence in substantially fewer iterations.

Discussion: we introduced SemDeDup, a simple yet tractable and effective method which leverages pre-trained embeddings to remove semantic duplicates which are highly semantically similar but not identical. Removing semantic duplicates improves learning speed and out-of-distribution performance while providing efficiency gains of up to 50% on the largely uncurated LAION and 15% on the partially curated C4. SemDeDup demonstrates the importance of data quality and the potential of data curation to dramatically improve training efficiency.

REFERENCES

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models, 2023. URL https://arxiv.org/abs/2301.03728.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/97af07a14cacba681feacf3012730892-Paper.pdf.

Andrei Broder. On the resemblance and containment of documents. 06 1997. doi: 10.1109/SEQUEN.1997.666900.

Kashyap Chitta, José M Álvarez, Elmar Haussmann, and Clément Farabet. Training data subset search with ensemble active learning. *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, 2021.

Won-Seok Choi, Dong-Sig Han, Hyundo Lee, Junseok Park, and Byoung-Tak Zhang. DUEL: Adaptive duplicate elimination on working memory for Self-Supervised learning. October 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Adv. Neural Inf. Process. Syst.*, 33:2881–2891, 2020.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5915–5922, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. DeepCore: A comprehensive library for coreset selection in deep learning. April 2022.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021b.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. Opt-iml: Scaling language model instruction meta learning through the lens of generalization, 2022. URL https://arxiv.org/abs/2212.12017.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models, 2022. URL https://arxiv.org/abs/2202.06539.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better, 2021a. URL https://arxiv.org/abs/2107.06499.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021b.

Yunxing Liao. *Dataset Deduplication with Datamodels*. PhD thesis, Massachusetts Institute of Technology, May 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Kristof Meding, Luca M. Schulze Buschoff, Robert Geirhos, and Felix A. Wichmann. Trivial or impossible—dichotomous data difficulty masks model differences (on ImageNet and beyond). In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=C_vsGwEIjAr.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Adv. Neural Inf. Process. Syst.*, 34, December 2021.

Jeff M Phillips. Coresets and sketches. January 2016.

Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training. *arXiv preprint arXiv:2301.02280*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021a.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021b.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro,

Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis amp; insights from training gopher, 2021. URL https://arxiv.org/abs/2112.11446.

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, 2019. URL https://arxiv.org/abs/1910.10683.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/recht19a.html.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryenvpEKDr.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022a.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022b.

Emily Silcock, Luca D'Amico-Wong, Jinglin Yang, and Melissa Dell. Noise-Robust De-Duplication at scale. December 2022.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *Neural Information Processing Systems (NeurIPS)*, June 2022.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, January 2016.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

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

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

A ADDITIONAL LAION METHODS

A.1 DATASETS AND TRAINING

The LAION dataset. To train large-scale multi-modal models, we used the LAION dataset (Schuhmann et al., 2022a), an open multi-modal dataset containing up to 5 billion image-text pairs scraped from the web. LAION data were filtered using a pre-trained CLIP model to only retain image-text pairs with an embedding similarity greater than 0.28. Image-text pairs containing very short captions or small images were also removed. A simple de-duplication method based on the image url was also performed.

The majority of our experiments were performed on the LAION-440M filtered subset of LAION-2B introduced by (Radenovic et al., 2023). This dataset was filtered according to three criteria: (1) high enough caption complexity; (2) the caption must contain an action; (3) any text present in the image cannot substantially overlap with the caption.

To ensure this CAT filtered LAION-440M subset did not impact our results, we also performed experiments on unfiltered data derived from LAION. While we attempted to collect the original LAION-400M subset (Schuhmann et al., 2021), unfortunately, much of the data were no longer available as the source data were removed from the web. We therefore used a reduced version of the LAION-400M subset containing the 233 million data points we were able to collect, which we call LAION-233M.

CLIP training. For CLIP training on LAION, we use the OpenCLIP implementation (Ilharco et al., 2021). All the models we trained have Vision Transformer Base (ViT-B-16) (Dosovitskiy et al., 2020) as an image encoder and Text Transformer Base (ViT-B-16) (Vaswani et al., 2017) as a text encoder. We trained all models with global batch size of 33k image-caption pairs. We fix the number of training epochs to 32 regardless of the dataset size. This results in training for a fewer number of iterations when training on deduplicated data, thereby achieving efficiency gains. We train with AdamW (Loshchilov & Hutter, 2017) and cosine learning rate schedule with warmup. The same peak learning rate of $5\text{x}10^{-4}$ was used for all models.

CLIP Evaluation For CLIP evaluation we use zero-shot evaluation on 31 different datasets. Table (A1) and Table (A2) in the Appendix list all the datasets we use for evaluation.

B SEMDEDUP ON NATURAL LANGUAGE

B.1 METHODS

We train language models on deduplicated versions the C4 dataset (Raffel et al., 2019). Since pre-training large language models on the entire C4 corpus is beyond our compute budget, we train on subsets of this data whose sizes are compute optimal given model size as per Hoffmann et al. (2022). We use the OPT model and training configurations (Zhang et al., 2022) to train 125M and 1.3B parameter models (see Table 1 in Zhang et al. (2022) for full specifications). We use the original number of warmup updates but adjust learning rate schedule such that all training runs anneal learning rate to 0 by the end of the training — this allows for fair comparisons of model performances across different dataset sizes. For 1.3B model size experiments, we increase the number of warmup updates to 5550 and reduce the peak learning rate to $6\text{x}10^{-5}$ to stabilize training.

We evaluate our trained language models on two independent validation sets: the validation text corpora used by OPT (Zhang et al., 2022) (referred to as "opt_valid") and a random sample of the instruction finetuning corpus used to train the OPT-IML family of models (Iyer et al., 2022), composed of verbalized prompts corresponding to a wide range of NLP tasks and their solutions (referred to as "prompts_with_answers").

To perform SemDeDup, we pass documents through the open-sourced pre-trained 125M OPT model (Zhang et al., 2022) and save the last layer embedding for the last token in the document. We then apply the same method described in Section D with $\mathcal{K} = 11000$ to cluster these embeddings. We compare to random pruning and the NearDup method described in Lee et al. (2021b). Note that the

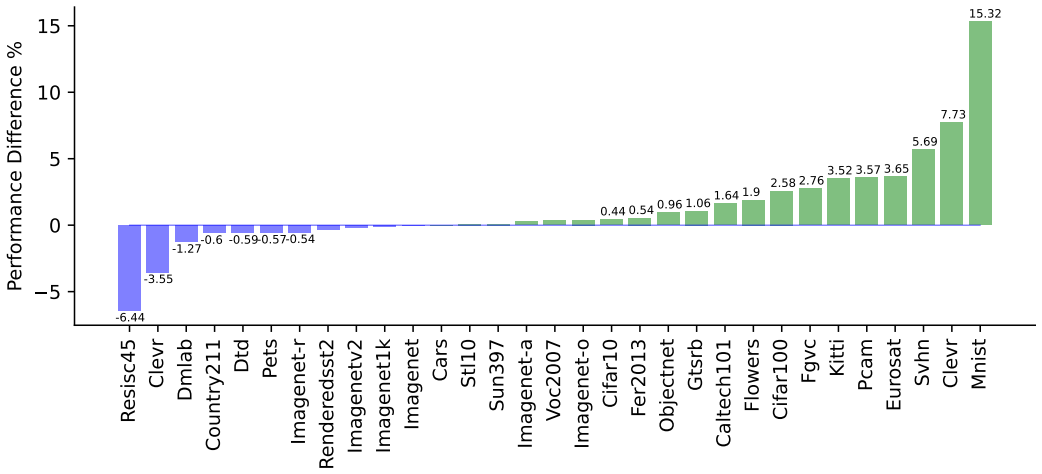


Figure A1: **SemDeDup improves zeroshot and OOD performance in many tasks with less pre-training.** A comparison of zeroshot evaluation performance between a baseline CLIP model (trained on 100% of the data) and our CLIP trained on 63% of LAION-440M after de-duplication on 31 tasks. The green bars show when SemDeDup outperforms the baseline model.

deduplication threshold values associated with different fractions of data remaining change compared to LAION-440M, as seen in Fig. A20.

B.2 RESULTS

In Fig. A2, we show the performance of SemDeDup versus random pruning. We observe that SemDeDup significantly outperforms random pruning as measured by perplexity on prompts_with_answers and average opt_valid performance. For a breakdown of performance on individual validation sets in opt_valid, see Figure A22 where we observe that SemDeDup beat random pruning on every single validation set in opt_valid.

Training on less data for one epoch naturally causes performance to decrease. Thus, we also explore whether continuing to train on the same smaller pruned datasets for more epochs will match the performance of a baseline model trained on a larger dataset. In Fig. A3, we train on datasets pruned with SemDeDup, but perform the same number of total training steps as the baseline model on the larger dataset (which was trained for 1 epoch). This causes the model to do multiple epochs over the pruned dataset. We observe that by training for multiple epochs over significantly pruned datasets we can reach the performance of a single-epoch run on the full dataset using 10-15% less compute. This is similar to the finding in Section 3.3. Notably, this efficiency gain is larger at higher pruning percentages, indicating that more aggressive pruning can yield more efficiency gains. This trend generally holds across the individual validation sets in opt_valid (see Figure A23).

On the C4 validation set, we observe that SemDeDup still outperforms random pruning in Figure A21. In Table A14 we compare SemDeDup to the NearDup baseline from Lee et al. (2021a). We observe that NearDup and SemDeDup have comparable performance as is expected, because with 4% pruning there is very little change to the underlying dataset.

B.3 WHAT IS BEING PRUNED IN LANGUAGE DATA?

In Figure A24 and Figure A25 we choose specific clusters and show a random sample of documents retained in the cluster after performing SemDeDup for different values of ϵ. In Figure A24, we observe that at low values of ϵ, we find semantic duplicates in the form of templated text, where typically few words (e.g. a geographic location or a name) is changed. This successfully evades exact-string deduplication methods but contains highly redundant information as seen in Figure A24. In Figure A25, we show an example of a cluster with semantically redundant duplicates — most

examples in this cluster are advertisements about Nike shoes. These examples are not necessarily templated text or have exact string matches, but are highly redundant nonetheless. We see in Figure A25 that at more aggressive pruning (i.e. higher ϵ) these semantically redundant duplicates get pruned. We note that exact string duplicates (i.e."perceptual duplicates for text") are rare since duplicate occurrences of any three-sentence spans were removed in C4 already.

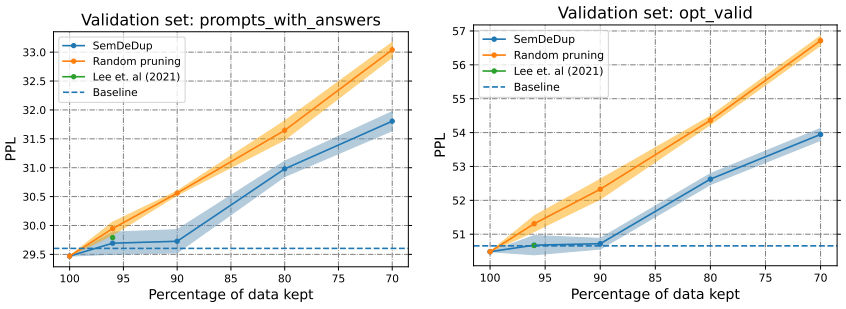


Figure A2: **SemDeDup applied to C4**. The x-axis corresponds to different percents of data kept, and the y-axis represents the perplexity on validation sets described in Section B.1 (lower is better). Each point is a separate 125M model trained on one-pass of its respective pruned dataset (mean and standard deviation across 3 random training seeds). The green point represents a 125M model trained on a version of C4 deduplicated via the NearDup method (Lee et al., 2021a). Note that NearDup (the single green point) keeps 96.1% of the data. SemDeDup can match this baseline performance while keeping only 90% of the data (see Table A16 for numerical comparison).

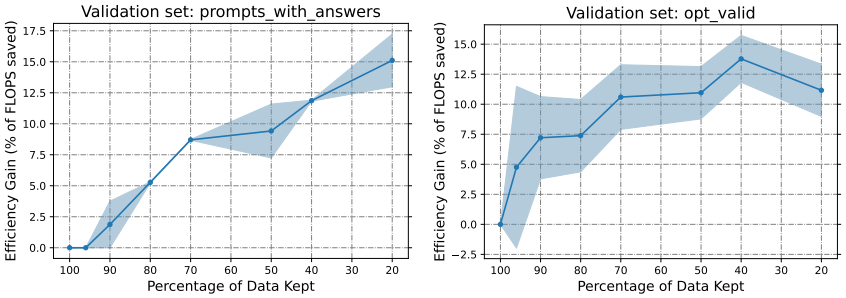


Figure A3: **SemDeDup allows compute efficiency gains by training on much smaller datasets for slightly longer.** We prune datasets via SemDeDup and continue training past one epoch until we reach baseline model perplexity. The x-axis is percentage of data kept, and the y-axis is the percentage of FLOPs saved. For example, training on the 80% pruned dataset reaches baseline model perplexity on prompts_with_answer in 95.0% of the baseline training, saving 5.0% compute. Mean and standard deviation provided across 3 random training seeds.

C Related Work

Much of the work in langauge and vision on deduplication has focused on the removal of exact duplicates. For example, Liao (2022) removed duplicates between the YFCC15M dataset (Thomee et al., 2016) and the ImageNet validation set to prevent train-test leakage. The C4 text corpus - used for training T5 (Raffel et al., 2019) - has been deduplicated by discarding repeated occurrences of any three-sentence spans. Lee et al. (2021a) showed that it's possible to further deduplicate this dataset without loss of performance by computing approximate n-gram overlap between documents using the MinHash technique (Broder, 1997). Rae et al. (2021) also applied MinHash based deduplication to curate training data for the Gopher model and demonstrated that training on the deduplicated dataset can result in lower perplexity across various validation sets. Kandpal et al. (2022) found that deduplication prevents memorization in LLMs and thus mitigates privacy concerns. More recent

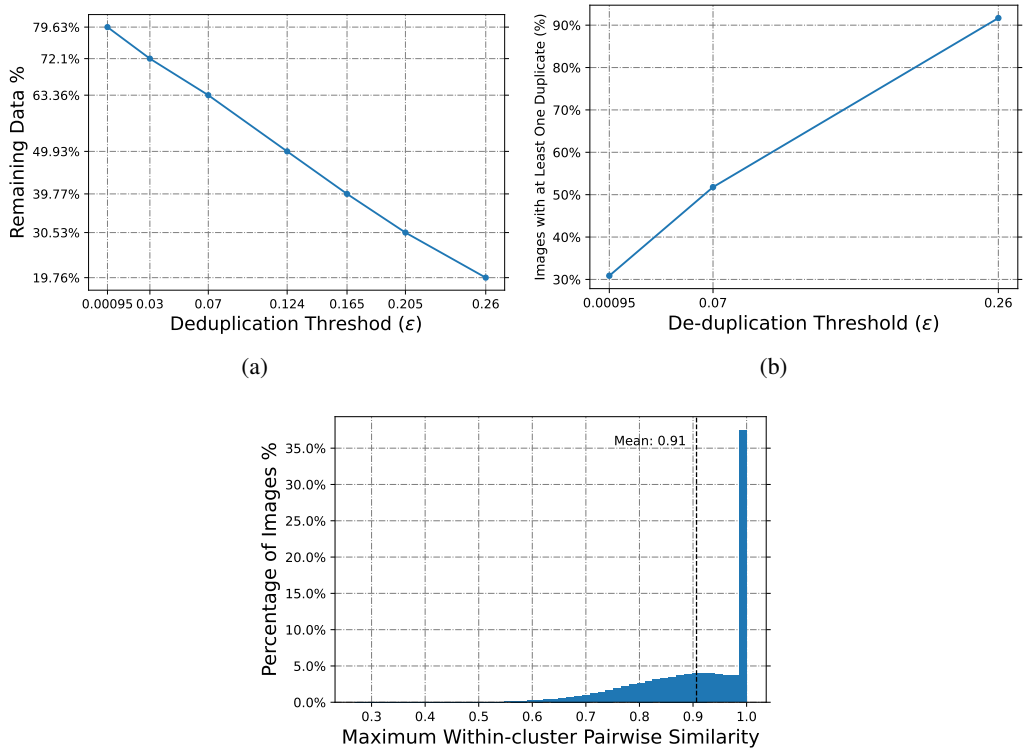


Figure A4: **Extreme semantic redundancy in LAION-440M.** (a) Fraction of data remaining as a function of deduplication threshold ϵ for LAION-440M. (b) Percentage of images in LAION-440M with at least one semantic duplicate as a function of ϵ. (c) Histogram of number of within-cluster image pairs in LAION-440M at a given cosine similarity.

works use forms of model-based feature extraction to improve the robustness of the similarity metric used for deduplication. Silcock et al. (2022) created a supervised dataset for detecting duplicate news articles and trained models to predict those labels. In the domain of computer vision, Choi et al. (2022) improves on SSL techniques by removing near-duplicates in some high dimensional feature space they learn.

Beyond deduplication, a host of classical machine learning approaches seek to achieve data efficiency by finding *coresets*, defined as small subsets of the training data that can be used to train a machine learning algorithm to the same test accuracy achievable when training on the entire training data (see e.g. (Guo et al., 2022; Phillips, 2016) for reviews). However, many coreset algorithms are computationally prohibitive and therefore are difficult to scale to web-scale data. In contrast to many traditional coreset algorithms, we develop an exceedingly simple and tractable algorithm that achieves both computational and data efficiency at scale.

Recent approaches to achieve data efficiency in deep learning have operated in a supervised setting by defining and finding "hard" examples not easily learned by partially or fully trained (ensembles of) models (Toneva et al., 2019; Paul et al., 2021; Chitta et al., 2021; Feldman & Zhang, 2020; Meding et al., 2022). Perhaps the closest to our work is a recent effort to break beyond neural power law scaling by pruning unlabelled data, using the embedding space of a pre-trained foundation model (Sorscher et al., 2022). However, the largest dataset for which these works examined data pruning was ImageNet. In contrast we move from relatively small, highly curated ImageNet scale to highly uncurated, web-scale data. Our analysis, at this new large and uncurated scale, reveals a possibly fundamental role for semantic deduplication as an important initial step in data-pruning for self-supervised learning that was not considered in prior data-pruning works.

D CLUSTERING TO REDUCE COMPUTATION

The time complexity of naive de-duplication is $\mathcal{O}(n^2)$ where n is the number of data points, making this approach impractical for large web-scale data. For example, the LAION-440M dataset would require $\approx 1.9\text{x}10^{17}$ similarity computations. The $\mathcal{K}$-means clustering step in SemDeDup reduces this complexity substantially from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^2/\mathcal{K})$ assuming approximately uniform cluster size. This means we only require $\approx 4.6\text{x}10^{12}$ intra-cluster comparisons instead of $\approx 1.9\text{x}10^{17}$ across all pairs, a 5-order of magnitude improvement.

D.1 WHAT DO SEMANTIC DUPLICATES LOOK LIKE?

Many semantic duplicates are of products which may have been displayed on multiple e-commerce websites, each with a slightly different style. As a result, semantic duplicates often contain different, but highly similar captions. While most clusters contained 20-40% duplicates, there are several remarkable outliers in redundancy in LAION-440M (Fig. A10), including one cluster containing $\approx 307,000$ copies of the European Union flag and another with $\approx 318,000$ copies of an icon of "Image not found."

At higher levels of ϵ in Fig. 2, A11, A12, and A13, we find fewer semantic duplicates, which are generally derived from the same source image, and more pairs which exhibit semantic redundancy instead, in which the same concept is present, but not derived from the same image source. For example, semantically redundant pairs may contain different images of similar objects or scenes.

D.2 WHY DO MODELS TRAINED ON UNCURATED DATA EXHIBIT SLOWER LEARNING?

We posit that successive learning iterations involving semantic duplicates yield redundant information, thereby wasting valuable computation on data points that are highly similar to those the model has already seen. By removing these semantic duplicates, we increase the fraction of data points which provide a marginal information gain to the model, thereby increasing learning speed (Sorscher et al., 2022).

E LAION-233M DE-DUPLICATION

To support our results on LAION-440M, we also de-duplicate a much smaller dataset of 233 millions images. We call this dataset LAION-233M. Usually, CLIP needs to be trained on more than 400 millions images as introduced in (Radford et al., 2021a), so de-duplicating LAION-233M is more challenging in this respect. We trained a baseline model on the 233 millions images and two models on 55% of the data, one on random subset and the other on de-duplicated subset. We trained all the models using the same hyperparameters we used for training on LAION-440M. We show ImageNet top1 zeroshot accuracy for these models in Fig. A5. The baseline model achieved 64.62% accuracy, while the SemDeDup model achieved 63.61% outperforming the model trained on random subset (61.3% accuracy).

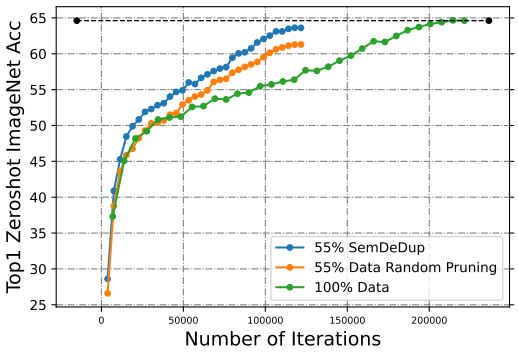


Figure A5: Training CLIP on 233 millions images from LAION-2B.

F CLIP ZEROSHOT EVALUATION

In this section we show the result of zeroshot evaluation for CLIP. Models trained on dataset de-duplicated SemDeDup outperform the baseline model in many tasks. In Table A1 we list the top1 zeroshot accuracy on 25 tasks and in Table (A2 we show the top1 zeroshot accuracy on 6 datasets for out-of-distribution robustness evaluation. Our complete evaluation set has 31 different datasets in total. When using only 63% of LAION-440M, SemDeDup outperforms the baseline model in 20 out of the 31 tasks. Fig. (A7) and Fig. (A8) show the performance of different models as a function on training dataset size.

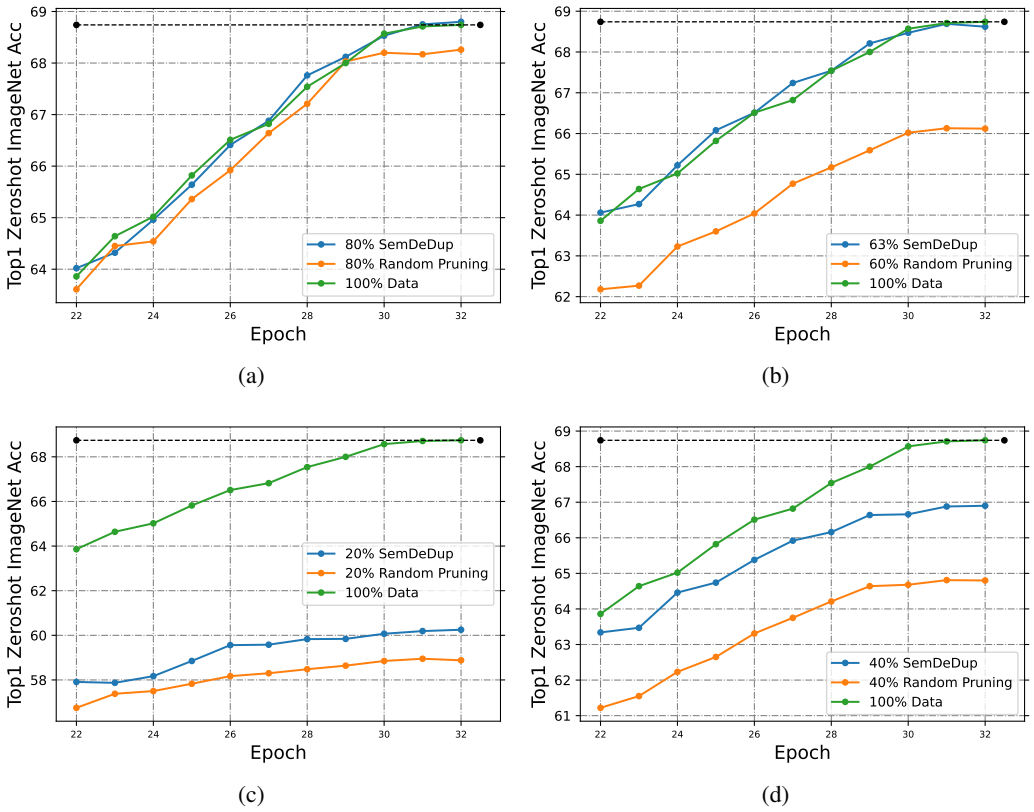


Figure A6: SemDeDup is always better than training on random subset from LAION-440M. The plot shows zeroshot top1 accuracy on ImageNet for CLIP models trained on different fractions of data .

G VISUALIZING EXAMPLES BEFORE AND AFTER DE-DUPLICATION

To visually show which images are removed by SemDeDup from LAION-440M dataset, we viualize some images from a random cluster before and after de-duplicaton. To do that, we choose a cluster randomly and sort its examples by the cosine similarity to the centroid. By doing this, we can show the similar images next to each other. Then we visualize a sequence of images before de-duplicaton. After that we run SemDeDup, remove examples and sort the remaining examples again. Finally we visualize the squence of images from the same indices we visualze them before de-duplication. Figures (A11-A13) show that after applying SemDeDup with different values for the de-duplication threshold ϵ, we keep the unique images. For example we in Fig. (A11) that the cluster contains many similar images of watches and as we remove more images with higher value of ϵ we keep more unique examples that are less semantically redundant.

Table A1: **Zeroshot evaluation** top1 accuracy on different datasets. Training CLIP on 63% of the data gives higher performance in 18/25 datasets.

Dataset / Model	dedup20	dedup40	dedup50	dedup63	dedup72	dedup80	baseline	rand80	rand60	rand40	rand20
Cars	63.33	78.14	80.26	81.43	82.05	**82.61**	81.42	80.96	79.26	77.74	71.57
Country211	14.16	17.74	18.26	18.44	18.70	18.20	**19.03**	18.39	16.75	15.97	12.88
Fer2013	41.72	47.90	46.21	**52.51**	48.94	45.03	51.96	43.37	43.42	42.27	42.67
Fgvc aircraft	4.44	11.49	12.42	**15.42**	15.27	15.09	12.66	14.31	13.11	9.27	8.85
GTSRB	38.22	37.20	36.22	43.06	41.00	35.74	42.00	**43.33**	41.88	25.72	32.28
Imagenet1k	60.24	66.90	68.27	68.66	**68.93**	68.80	68.74	68.29	66.12	64.82	58.86
MNIST	44.29	31.87	22.93	48.55	42.75	**48.86**	33.23	43.82	35.73	36.32	19.22
Renderedsst2	51.46	53.65	52.72	50.80	52.99	**57.17**	51.13	52.72	51.29	52.22	45.47
STL10	96.06	96.85	97.50	**97.71**	97.69	97.21	97.62	97.49	97.38	97.08	94.31
SUN397	64.81	67.98	68.26	68.89	69.25	**69.76**	68.79	69.08	67.96	65.51	60.76
VOC2007	77.94	79.51	79.74	**80.37**	79.75	78.61	80.01	77.97	79.43	77.96	74.42
Caltech101	83.05	84.40	84.98	**85.06**	84.35	84.75	83.42	83.69	83.93	83.38	80.62
CIFAR100	72.09	75.71	75.17	**77.19**	77.16	77.08	74.61	76.02	74.08	72.37	67.79
CIFAR10	92.80	93.78	94.01	94.00	**94.49**	94.13	93.56	94.25	93.95	92.68	89.14
Clevr object dist	15.75	**23.05**	15.75	19.48	21.95	21.82	23.03	18.45	15.59	18.60	16.21
Clevr count all	25.37	26.36	30.85	31.87	**34.73**	20.31	24.14	15.37	26.43	14.85	21.67
DMLAB	13.16	17.62	17.99	19.20	18.52	20.23	20.46	18.50	**21.05**	17.12	19.36
DTD	49.73	53.51	54.31	56.76	**58.94**	57.66	57.34	57.02	53.35	50.96	41.76
Eurosat	44.07	51.28	51.70	59.46	57.02	59.72	55.81	**59.81**	48.63	51.26	50.00
Flowers	45.21	62.21	67.67	69.78	**70.48**	66.29	67.88	68.39	65.43	62.42	58.16
Kitti vehicle dist	20.39	13.36	14.35	14.77	19.97	**26.72**	11.25	11.11	20.68	17.02	11.25
PCAM	49.69	48.83	47.62	**52.66**	50.09	52.14	49.09	50.11	41.28	55.02	56.59
Pets	77.87	87.30	89.72	90.02	90.16	90.57	**90.60**	90.49	89.86	88.72	82.50
Resisc45	46.76	57.56	51.69	51.49	50.14	53.57	**57.93**	54.06	51.65	49.29	46.72
SVHN	34.80	33.64	26.24	**40.87**	33.96	32.68	35.18	26.77	26.67	32.70	25.26

Table A2: **Out-of-distribution Robustness:** for CLIP model we trained on different number of examples. The two models trained on 63% and 72% of LAION440M with our de-duplication method have higher average accuracy over 6 datasets.

Model/Dataset	ImageNet-A	ImageNet-O	ImageNet-R	ImageNet_Sketch	ImageNetV2	ObjectNet	Average
dedup20	31.35	52.25	72.69	46.98	52.71	51.0	51.16
dedup40	38.73	49.3	77.08	51.93	59.21	54.98	55.21
dedup50	39.68	48.55	77.74	53.54	60.37	55.36	55.87
dedup63	39.07	48.45	78.24	53.86	60.56	56.33	**56.08**
dedup72	39.53	47.6	78.61	53.7	61.23	56.28	**56.16**
dedup80	39.12	47.95	78.53	53.82	60.59	54.72	55.79
baseline100	38.79	48.05	78.77	53.91	60.77	55.36	55.94
rand80	37.87	47.7	78.04	52.81	60.02	54.3	55.12
rand60	34.6	47.5	75.61	51.18	57.97	53.22	53.35
rand40	31.88	49.1	73.65	49.02	56.83	49.57	51.67
rand20	23.43	49.4	66.74	43.76	50.67	43.57	46.26

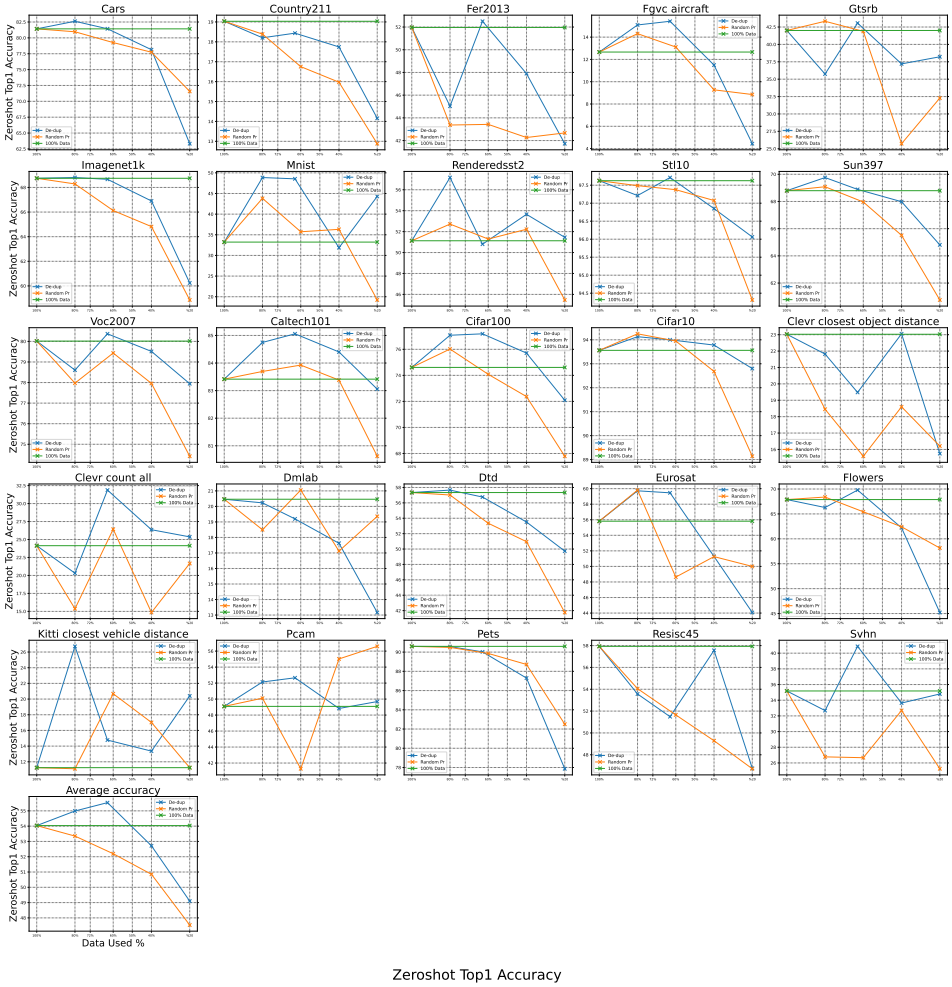


Figure A7: Zeroshot performance on 25 datasets. The last plot shows the average performance over all datasets.

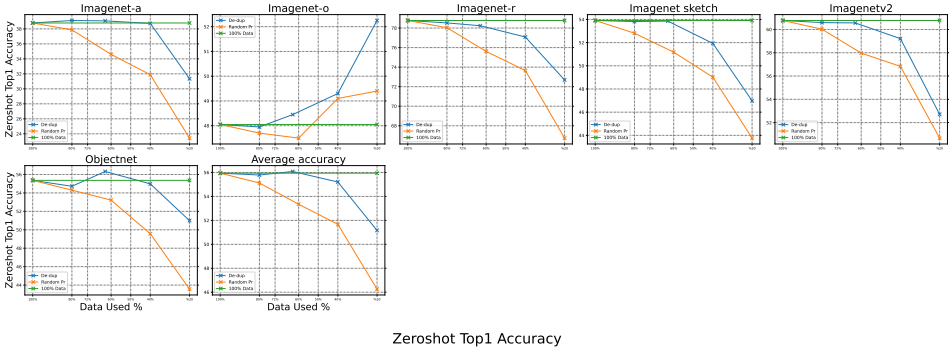


Figure A8: Out-of-distribution zeroshot performance zeroshot performance on 6 datasets. The last plot shows the average performance over all datasets.

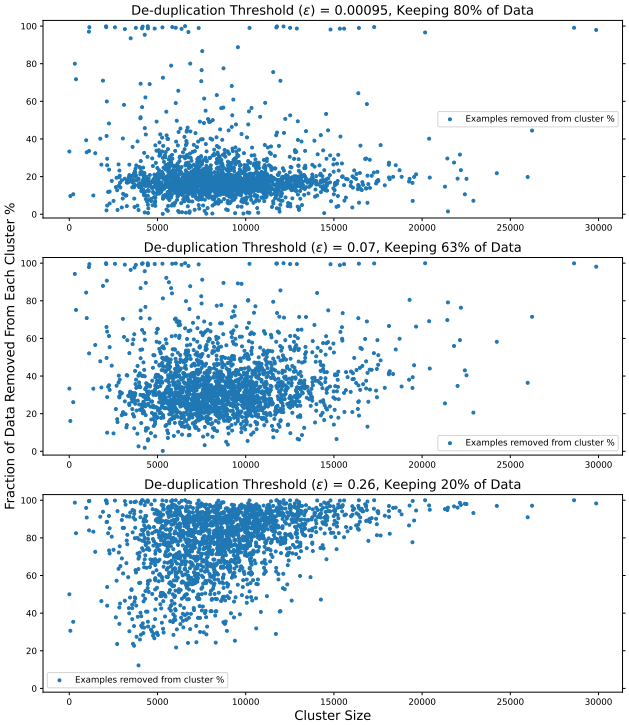


Figure A9: How many images can we remove from each cluster?. Moving from top to down we increase ϵ value. As we increase ϵ, more examples are removed from each cluster. The x axis corresponds to the cluster size. We notice that most of examples from the large clusters (the points to the right) are removed when ϵ becomes large. The points in this figure are for 2000 clusters sampled randomly from a total of 50,000 clusters.

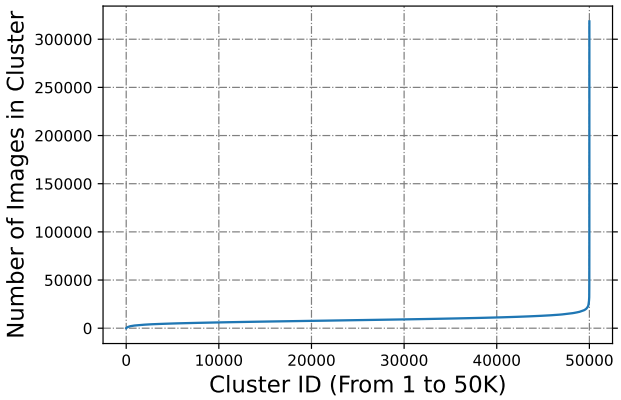


Figure A10: The number of images in each cluster for 50000 clusters of LAION-440M images after running K-means clustering on images embeddings. The average cluster size is 8,748, but we also see few clsuters with more than 300,000 examples.

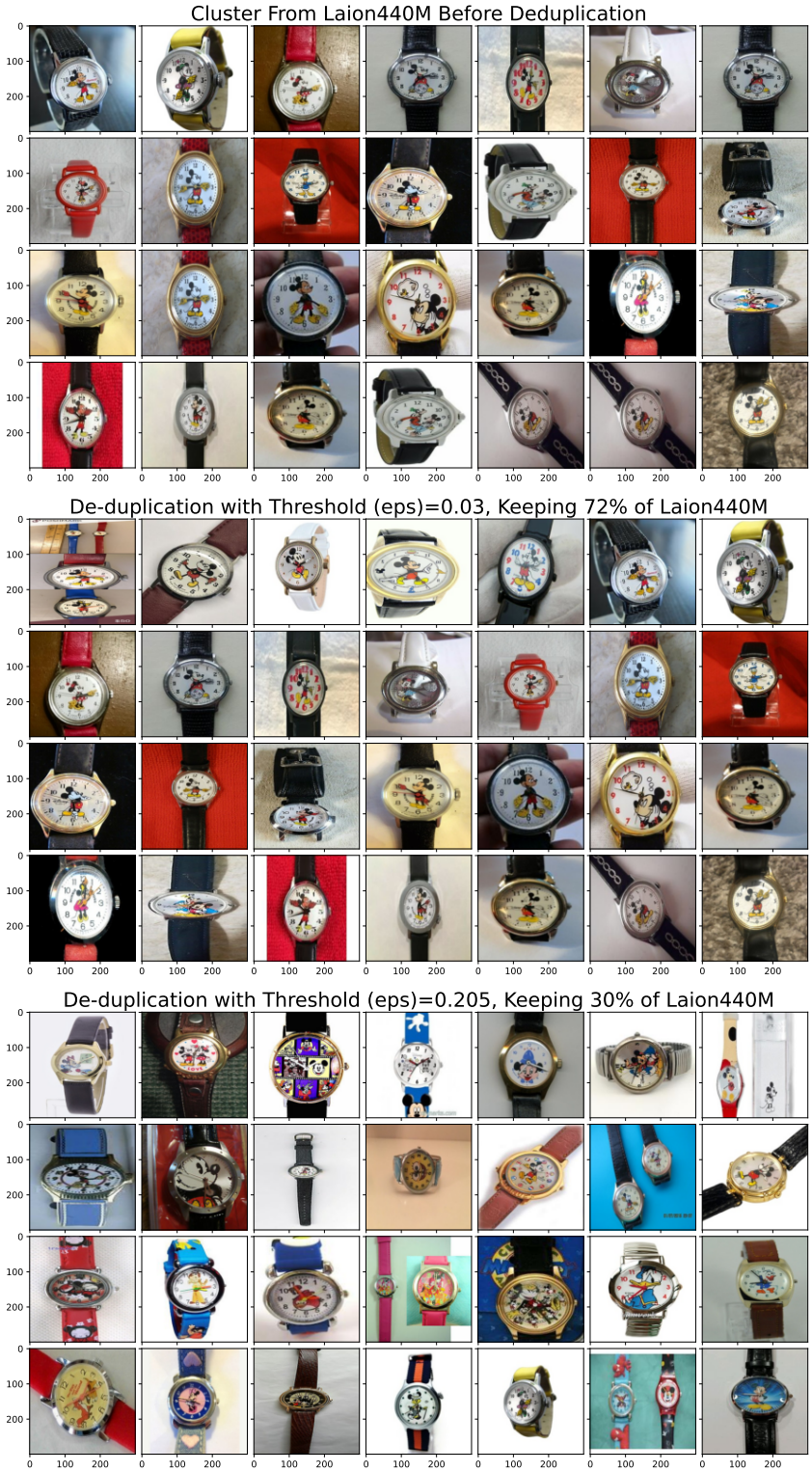


Figure A11: Examples from the same cluster from LAION-440M dataset before and after de-duplication. Images are sorted by the cosine similarity to the cluster centroid. As we increase the de-duplication threshold we start to see more unique images.

Figure A12: Examples from the same cluster from LAION-440M dataset before and after de-duplication. Images are sorted by the cosine similarity to the cluster centroid. As we increase the de-dplication threshold we start to see more unique images.

Figure A13: Examples from the same cluster from LAION-440M dataset before and after de-duplication. Images are sorted by the cosine similarity to the cluster centroid. As we increase the de-dplication threshold we start to see more unique images.

validation set	Baseline Training (no pruning)	NearDup from Lee et al. (2021a)	Random	SemDeDup
C4	38.95 +/- 0.07	39.46 +/- 0.14	39.51 +/- 0.07	**39.35** +/- 0.16
opt_valid	50.66 +/- 0.19	**50.67** +/- 0.03	51.31 +/- 0.23	**50.68** +/- 0.28
prompts_with_answers	29.60 +/- 0.15	29.79 +/- 0.11	29.95 +/- 0.11	**29.69** +/- 0.19

Figure A14: Comparison of perplexity values for 125M after pruning via different methods at 96% pruning. Note that Lee et al. (2021a) pruned 3.9 % of examples, while above Random and SemDeDup prune 4% of examples. Mean and standard deviation provided across 3 training seeds.

validation set	Baseline Training (no pruning)	Random	SemDeDup
C4	38.95 +/- 0.07	40.43 +/- 0.06	**39.80** +/- 0.40
opt_valid	50.66 +/- 0.19	52.33 +/- 0.29	**50.72** +/- 0.16
prompts_with_answers	29.60 +/- 0.15	30.56 +/- 0.04	**29.73** +/- 0.20

Figure A15: Comparison of perplexity values for 125M after pruning via different methods at 90% pruning. Mean and standard deviation provided across 3 training seeds.

validation set	Baseline Training (no pruning)	Random	SemDeDup
C4	38.95 +/- 0.07	42.16 +/- 0.03	**41.98** +/- 0.09
opt_valid	50.66 +/- 0.19	54.36 +/- 0.12	**52.63** +/- 0.16
prompts_with_answers	29.60 +/- 0.15	31.65 +/- 0.16	**30.98** +/- 0.13

Figure A16: Comparison of perplexity values for 125M after pruning via different methods at 80% pruning. Mean and standard deviation provided across 3 training seeds.

validation set	Baseline Training (no pruning)	NearDup from Lee et al. (2021a)	Random	SemDeDup
C4	46.16	46.85	**46.15**	46.56
opt_valid	59.64	59.27	59.20	**58.89**
prompts_with_answers	34.04	33.93	33.91	**33.83**

Figure A17: Comparison of perplexity values for 1.3b model after pruning via different methods at 96% pruning. Note that Lee et al. (2021a) pruned 3.9 % of examples, while above Random and SemDeDup prune 4% of examples. Due to compute restrictions we do not provide random seed standard deviations.

validation set	Baseline Training (no pruning)	Random	SemDeDup
C4	46.16	**62.50**	63.36
opt_valid	59.64	76.09	**75.57**
prompts_with_answers	34.04	43.54	**42.33**

Figure A18: Comparison of perplexity values for 1.3b model after pruning via different methods at 50% pruning. Due to compute restrictions we do not provide random seed standard deviations.

validation set	Baseline Training (no pruning)	Random	SemDeDup
C4	46.16	303.71	**108.95**
opt_valid	59.64	347.77	**115.69**
prompts_with_answers	34.04	269.96	**72.64**

Figure A19: Comparison of perplexity values for 1.3b model after pruning via different methods at 20% pruning. Due to compute restrictions we do not provide random seed standard deviations.

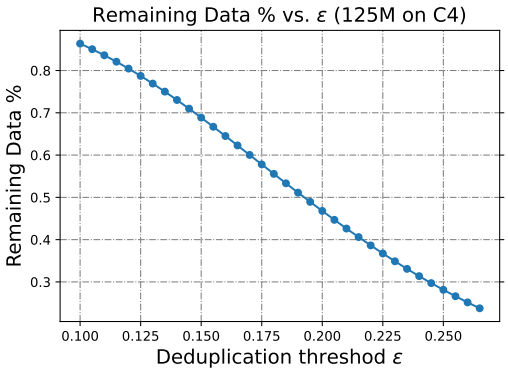
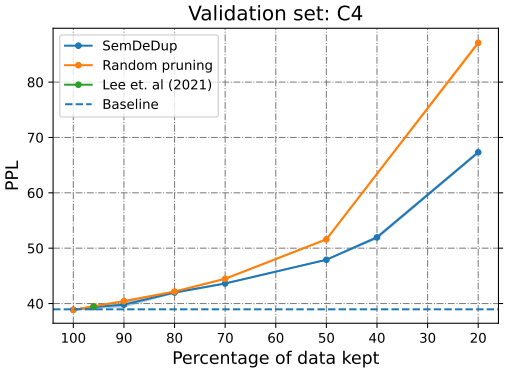
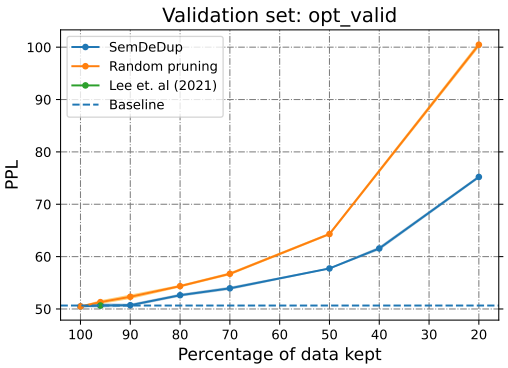


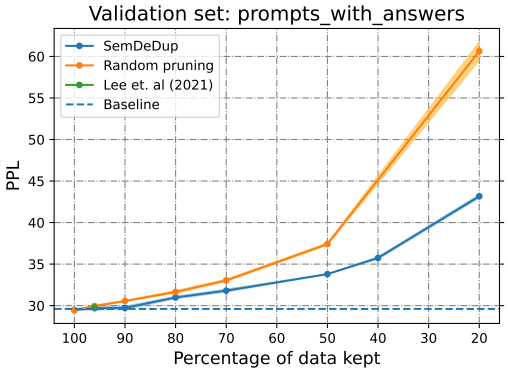
Figure A20: **Percent Data Remaining versus ϵ for C4**. The x-axis corresponds to different values of ϵ from Section 2, the y-axis represents the corresponding fraction of data in our subset of C4.



(a)



(b)



(c)

Figure A21: SemDeDup performance at different fractions of data for the 125M OPT model. We show results for the C4 validfation set (top left), opt_valid (top right) and prompts_with_answers (bottoms). These are the same graphs as Figure A2, but for a wider range of percentage of data kept. We note that SemDeDup consistently outperforms random pruning at lower percentages of data kept.

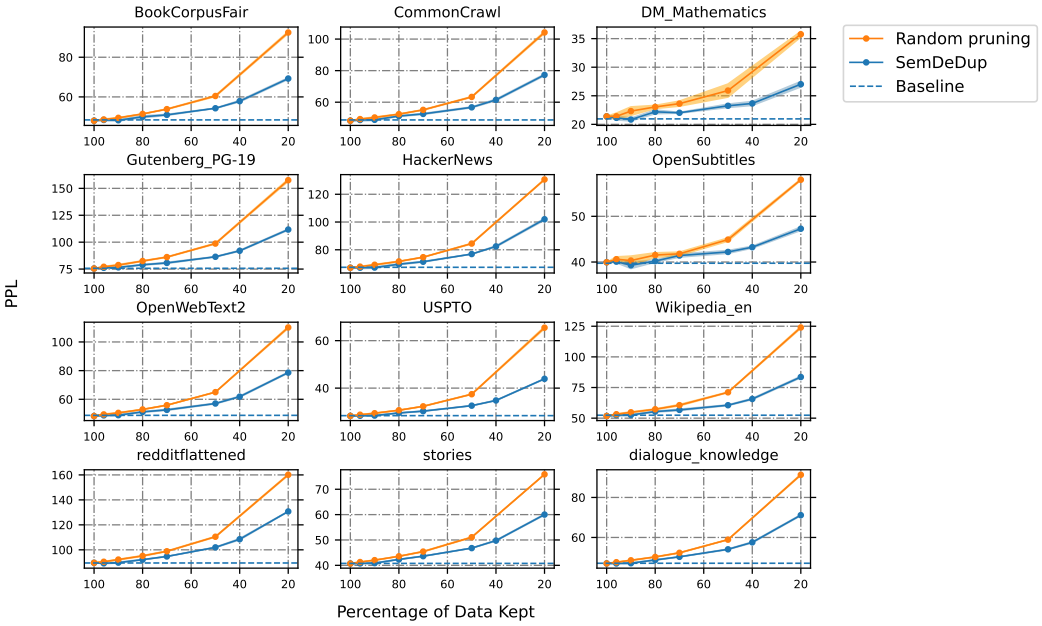


Figure A22: Percentage of Data Kept vs. Perplexity on individual validation sets within opt_valid. Runs are averages across 3 training seeds, and shaded regions represents 1 standard deviation from the mean. The title of each plot represents the name of the individual validation set within opt_valid. Note that on all tasks, SemDedup significantly random pruning, especially at low percentages of data kept.

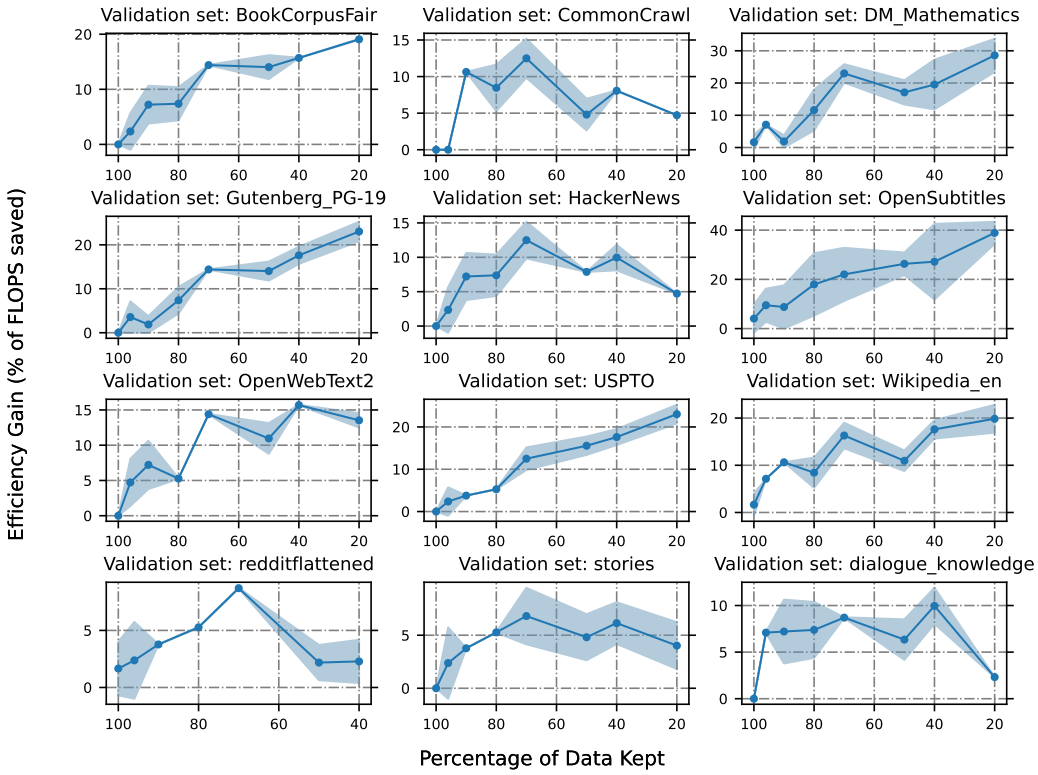


Figure A23: Percentage of Data Kept vs. Efficiency Gain on individual validation sets within opt_valid. Runs are averaged across training seeds where the model achieves baseline perplexity at some point in training, and shaded regions represents 1 standard deviation from the mean. The title of each plot represents the name of the individual validation set within opt_valid.

Keeping 90% data
text
It appears that you already have an account on this site associated with . To connect your existing account...

Keeping 100% data (i.e. no pruning)
text
It appears that you already have an account on this site associated with . To connect your existing account... You are visiting the placeholder page for Wells Williams. This page is here because someone used our ... You are visiting the placeholder page for Mathew Barrett. This page is here because someone used our ... You are visiting the placeholder page for Marcus Slatar. This page is here because someone used our ... You are visiting the placeholder page for Bernice Andrews. This page is here because someone used our ... You are visiting the placeholder page for Emiko Chille. This page is here because someone used our ... You are visiting the placeholder page for Landon Buckland. This page is here because someone used our ... . You are visiting the placeholder page for Kylie Dickens. This page is here because someone used our ...

Figure A24: Example of semantic de-duplication with SemDeDup (cluster 4500)

Keeping 20% data
text
cheap jordan shoes from china free shipping,order maroon foams , jordan blue retro 12 , jordans sz 10 , all white 14s... Booming business thanks to Cristiano Ronaldo! Nike Presents Cristiano Ronaldo – CR7 Winter Collection ...

Keeping 90% data
text
Purchase from us, you can get max discount and free shipping.Free shipping and returns on Nike Jordans ....
Product range. Adidas collections are divided into three groups: Sport Performance, Originals, and Sport Style ...
cool jordans for boys , foamposite paranorman , new black and white foams , lebron 1's ,cheap jordans online ...
This Comfortable Nike Huarache Free Basketball And Running has 1600 x 900 pixel resolution with jpeg format. ..
Top Rating: "Best high performance product." Performance efficiency. That is the motto of our textile engineers by...
cheap jordan shoes online free shipping order cheap jordans for sale free shipping. Air Jordan 1's new theme ...
...
Trendy Men's Nike Kyrie 1 Best Seller 'All Star' Multicolor at high discount. Buy Nike Trainers - The Kyrie 1 All ...

Figure A25: Example of semantically redundant de-duplication with SemDeDup (cluster 4900)