Data-Unlearn-Bench: Making Evaluating Data Unlearning Easy

Abstract

Evaluating machine unlearning methods remains technically challenging, with recent benchmarks requiring complex setups and significant engineering overhead. We introduce a unified and extensible benchmarking suite that simplifies the evaluation of unlearning algorithms using the KLoM (KL divergence of Margins) metric (GRP^{+}24). Our framework provides precomputed model ensembles, oracle outputs, and streamlined infrastructure for running evaluations out of the box. By standardizing setup and metrics, it enables reproducible, scalable, and fair comparison across unlearning methods. We aim for this benchmark to serve as a practical foundation for accelerating research and promoting best practices in machine unlearning. Our code and data are publicly available.

1. Introduction

The growing reliance on machine learning in sensitive and regulated domains, such as healthcare and finance, has raised critical concerns regarding the ability of models to selectively forget training data upon request. This process is known as *machine unlearning*. Effective machine unlearning ensures that an unlearned model trained with certain data behaves as though specific data (the *forget set*) was never included in the training set. Although retraining from scratch without the unwanted data offers a theoretically perfect solution, it remains computationally impractical, particularly for large-scale deep learning models.

Recent literature has introduced various heuristic methods aimed at approximate unlearning, yet rigorously evaluating these methods remains challenging. A significant barrier is the substantial computational cost involved in hyperparameter searches and repeated retrainings necessary to obtain reliable evaluation results. Current benchmarks are often complex, opaque, or costly, leading researchers to either spend extensive resources on evaluations or rely on oversimplified heuristics. This complexity can inadvertently promote evaluation methods that are susceptible to ”gaming,” where improved scores do not necessarily reflect genuine unlearning efficacy.

Motivated by these challenges, we propose a unified benchmarking framework that simplifies the evaluation of data unlearning methods. Our framework provides standardized infrastructure and readily available resources, including precomputed model ensembles, oracle outputs, and established evaluation protocols. Central to our evaluation is the KLoM (KL divergence of Margins) metric, which quantifies the similarity of an unlearned model's predictive margins to those of an oracle model retrained without the target data (GRP^{+}24). By offering these resources publicly, our goal is to significantly lower the computational overhead of evaluations and to encourage the development of efficient heuristic approximations for unlearning metrics.

Beyond immediate computational efficiency, our benchmark also facilitates deeper investigation into fundamental aspects of machine unlearning, such as scaling laws related to model sizes and the transferability of unlearning across different data subsets. Furthermore, we outline future extensions, such as incorporating complementary evaluation metrics like the Gaussian Unlearning Score (GUS) (PDL^{+}24) and addressing limitations observed in existing synthetic forgetting tasks such as TOFU (MFS^{+}24).

Through standardized, reproducible, and efficient evaluation of unlearning methods, we hope our benchmark accelerates progress towards practical, reliable, and computationally efficient unlearning methods, supporting more robust and responsible machine learning.

Scope. While our benchmark focuses on classification models evaluated via predictive margins under the logistic loss, its insights extend to generative models. These models, including large language models (LLMs), are trained with cross-entropy loss, which decomposes into conditional classification tasks. The evaluation methods we propose are conceptually aligned with the unlearning challenges in generative modeling. Our work provides tools that can support future efforts to evaluate unlearning in generative AI systems.

Acknowledgements. This benchmark builds on tools, methodology, and insights first developed in the *Attribute to Delete* study (GRP^{+}24), which laid the groundwork for using margin-based divergences as a principled evaluation framework for machine unlearning. We thank the authors for their foundational contributions.

2. Metrics

2.1. Problems with Existing Metrics: U-LiRA

We recall that the goal of data machine unlearning is to produce models whose behavior closely matches that of models retrained without the forget set. Achieving this requires statistical closeness between the distributions of the unlearned model and an oracle model, defined as the model retrained from scratch without the forget set. However, directly evaluating this objective has proven challenging, prompting the development of empirical proxies like U-LiRA (HST$^+$24), which assess unlearning quality via adversarial distinguishability.

U-LiRA leverages membership inference attacks (CCN$^+$22) to measure if an adversary can discern whether a data point originated from the forget set or a held-out validation set based solely on the model's output. Ideally, responses from the unlearned model to both sets would be indistinguishable, implying that an adversary cannot perform better than random guessing.

Despite its intuitive appeal, U-LiRA has several limitations. First, it is susceptible to manipulation. A trivial strategy to achieve a perfect U-LiRA score involves outputting constant margins across all inputs, rendering outputs indistinguishable. While this guarantees perfect indistinguishability, it completely eliminates model utility. This highlights a structural vulnerability in U-LiRA: it prioritizes indistinguishability without enough enforcing of utility constraints. Consequently, algorithms optimized for U-LiRA might inadvertently collapse model functionality, misleadingly inflating performance metrics.

Second, U-LiRA does not align fully with the formal definition of unlearning. The (ε, δ)-unlearning criterion demands that the full distribution of model outputs, not merely distinguishability at specific points, must closely match the oracle distribution.

Another issue with U-LiRA is treating the forget set itself. The metric evaluates performance by randomly generating multiple forget sets and comparing models that unlearned specific points against models that unlearned other points. While this approach captures some variations, it overlooks potential "compound effects" arising from the particular composition of a given forget set. Specifically, the impact of removing one point may depend critically on the removal of other points simultaneously. Such nuanced interactions are obscured when forget sets vary randomly across evaluations. In contrast, methods like KLoM address this limitation by fixing the forget set, thus providing a clearer understanding of compound interactions among points.

In summary, although U-LiRA provides a useful heuristic for evaluating unlearning, its susceptibility to manipulations and its scope limitations relative to the formal definition of unlearning indicate that caution is necessary when interpreting results. Metrics that directly estimate distributional divergence from oracle models inherently consider utility and are better suited for rigorous evaluation of unlearning quality.

2.2. KLoM

The KL divergence of margins (GRP$^+$24) (KLoM) is a metric for empirically assessing machine unlearning. It measures how similar the output distributions of unlearned models are to those of oracle models, which are retrained from scratch without the forget set. KLoM implements a relaxed version of the standard (ε, δ) unlearning definition by substituting KL divergence for approximate max divergence and by comparing model outputs instead of parameters.

To compute KLoM, we first define the margin. For an input x with true label y_x, let $f(x;\theta) \in \mathbb{R}^K$ be the output logits produced by a model θ. The margin of model θ on input x is defined as:

$$\varphi(x;\theta) = (f(x;\theta))_{y_x} - \log \sum_{k \neq y_x} \exp((f(x;\theta))_k).$$

This logit-gap formulation captures how separable the correct prediction is from alternatives and is both numerically stable and aligned with our implementation.

We then compare margin distributions from (i) oracle models $\{\theta_i^o\}_{i=1}^N$ (trained on the retain set $S \setminus F$) and (ii) unlearned models $\{\theta_i^f\}_{i=1}^N$ (obtained by applying unlearning algorithm U to models originally trained on the full dataset S). For each data point x, we compute its margin under each oracle and unlearned model, yielding two empirical margin distributions: $\{\varphi(x;\theta_i^o)\}_{i=1}^N$ and $\{\varphi(x;\theta_i^f)\}_{i=1}^N$. These are binned into histograms, and their KL divergence yields the pointwise metric:

$$\text{KLoM}(x) = D_{\text{KL}}(\text{Hist}(\{\varphi(x;\theta_i^o)\}_{i=1}^N) \,\|\, \text{Hist}(\{\varphi(x;\theta_i^f)\}_{i=1}^N))$$

which can be averaged or aggregated across points in the forget, retain, or validation sets. Figure 1 provides a visual overview of this process.

Assumptions and Robustness KLoM assumes only that we can draw repeated samples from the oracle and unlearned model distributions and observe their outputs. Unlike other metrics such as U-LiRA, it does not rely on Gaussian approximations or adversarial distinguishers. It is robust to "gaming" by degenerate unlearning strategies (e.g., returning a random model), which are penalized by KL divergence against the oracle distribution. This makes KLoM harder to fool than membership-inference-based metrics.

Hyperparameters and Practical Estimation KLoM depends on several hyperparameters: (i) the number of models

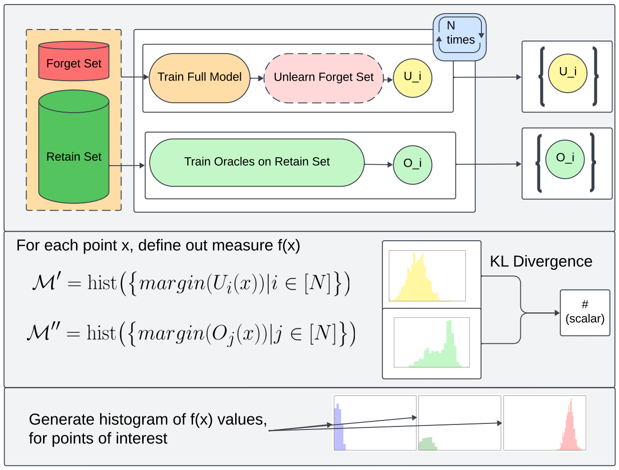

Forget Set
Retain Set
Train Full Model
Unlearn Forget Set
U_i
N times
Train Oracles on Retain Set
O_i
U_i
O_i
For each point x, define out measure f(x)
$\mathcal{M}' = \mathrm{hist}\big(\{margin(U_i(x)) | i \in [N]\}\big)$
$\mathcal{M}'' = \mathrm{hist}\big(\{margin(O_j(x)) | j \in [N]\}\big)$
KL Divergence
# (scalar)
Generate histogram of f(x) values, for points of interest

Figure 1: Overview of the KLoM methodology (GRP+24).

N, set to 100 oracle and 100 unlearned models for stable estimation; (ii) a clipping range of $[-100, 100]$ to suppress extreme margin values from unstable models; (iii) the number of histogram bins, fixed at 20 to balance resolution and variance; and (iv) a smoothing constant $\varepsilon = 10^{-5}$ to prevent empty histogram bins, which limits KLoM values to a maximum of approximately 12. We find the default parameter selection to be suitable for a fair evaluation across methods. In Appendix A.1 we show KLoM scores versus the number of compared models N and conclude that 100 models is enough for a reliable evaluation.

2.3. Future Directions: Extending to LLMs

Some current LLM unlearning evaluations for unlearning, such as TOFU (MFS+24) and Weapons of Mass Destruction Proxy (WMDP) benchmark (LPG+24), rely on multiple-choice question (MCQ) answering. While very convenient, this format fails to test whether the model has truly forgotten information. For instance, a model can learn to detect sensitive topics and produce generic or misleading outputs without necessarily removing the underlying knowledge (QPL+24). This evades detection while preserving the data internally. Robust evaluation should go beyond surface accuracy and measure whether the model's output distribution matches that of a retrained oracle. Without this, current methods risk overstating unlearning success.

Teacher-forcing KLoM We propose to extend KLoM to language models by evaluating margin divergence under teacher forcing. For a token sequence $x = (w_1, \ldots, w_T)$ and a model θ, we compute the margin $\varphi_t(x; \theta)$ at each prediction step t. This margin represents the model's confidence in the true next token w_{t+1} (from sequence x) relative to alternatives, given the prefix $x_{<t+1} = (w_1, \ldots, w_t)$. It is computed using the logit-gap definition analogous to $\varphi(x'; \theta)$ in Section 2.2, where x' corresponds to the input context (prefix $x_{<t+1}$) and the 'true label' is w_{t+1}.

At each prediction step t, we compare margin histograms from oracle $\{\theta_i^o\}_{i=1}^N$ and unlearned $\{\theta_i^f\}_{i=1}^N$ model ensembles. For a sequence x, we generate margin sets $\{\varphi_t(x; \theta_i^o)\}_{i=1}^N$ and $\{\varphi_t(x; \theta_i^f)\}_{i=1}^N$. Let $\mathrm{Hist}_t^o(x)$ and $\mathrm{Hist}_t^f(x)$ be histograms from these respective sets. Then $KLoM_t(x) = D_{\mathrm{KL}}(\mathrm{Hist}_t^o(x) \,\|\, \mathrm{Hist}_t^f(x))$. The teacher-forcing KLoM is the average across token positions, $\overline{KLoM}(x) = \frac{1}{T}\sum_{t=1}^{T} KLoM_t(x)$, aggregated over a dataset $\mathcal{D}$ as $\overline{KLoM}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{x \in \mathcal{D}} \overline{KLoM}(x)$.

This formulation preserves the original metric's robustness while enabling distributional comparison for next-token predictions in autoregressive models. It requires no changes to KLoM hyperparameters and is fully compatible with standard teacher-forcing evaluation.

Crucially, teacher-forcing KLoM is significantly harder to game than multiple-choice formats. Rather than checking if a model avoids specific outputs, it compares the full predictive distribution against that of a ground truth model, capturing subtler forms of retained knowledge.

3. Benchmark

Evaluating machine unlearning requires comparing the predictions of unlearned models to those of oracle models retrained without the forget set. Doing so reliably demands generating ensembles of pre-trained and oracle models, computing classification margins for each, and then estimating divergence metrics such as KLoM. Establishing such a reliable evaluation from scratch induces substantial overhead: for each forget set, one would typically need to train N full-data models and N oracles on the retain set, and extract per-example margin distributions from both. The cost of this setup is significant, both computationally and in engineering effort, presenting a bottleneck for rapid development and fair comparison of new methods.

To address this, our benchmark is designed with three key principles: (i) *Reusable infrastructure:* Pre-trained and oracle model ensembles are agnostic to the unlearning method under test and are expensive to compute. We precompute and distribute these components, allowing users to focus solely on the unlearning algorithm. (ii) *Standardized evaluation:* The benchmark provides tested implementations of core evaluation routines, reducing the risk of methodological errors and improving reproducibility. Users can trust that results are measured under consistent conditions. (iii) *Turnkey experimentation:* A complete experimental pipeline supports YAML-based configuration, automatic path resolution, checkpointing, and seamless scheduling of large-scale hyperparameter sweeps. Implementing a new method typically requires only a short function definition, while the infrastructure handles training, evaluation, and logging.

Included Datasets and Resources We provide full experimental support for two used benchmarks (GRP+24). For CIFAR-10 (Kri09) with ResNet-9 models we provide 100+ pre-trained models trained on the full dataset, 10 distinct forget sets, each with 100+ corresponding oracle models trained on the retain set, precomputed classification margins for all models and evaluation sets. For Living-17 (STM21; DDS+09) with ResNet-18 (HZRS16) models. We include: 100+ full-data pre-trained models, 3 forget sets, each with 100+ corresponding oracle models, precomputed margins for oracle and pre-trained ensembles.

Quick Start The benchmark supports a high-level workflow that requires no manual script editing or path configuration. All experiments are specified via YAML configuration files and launched via a generated script. For example, to evaluate gradient ascent on CIFAR-10 on a predefined hyperparameter grid (GRP+24) (learning rates $\{10^{-5}, 10^{-3}, 10^{-2}\}$; epochs $\{1, 3, 5, 7, 10\}$):

```
# Step 1: Generate config files
python config.py
```

This creates files like:

```
config/unlearning_method-ascent_forget_\
dataset-cifar10_epochs-[1,3,5,7,10]_\
forget_id-1_lr-1e-05_model-resnet9_\
optimizer-sgd_N-100_batch_size-64.yml
```

Then we can easily prepare a multi-gpu launch that will schedule and execute our experiments

```
# Step 2: Create a GPU launch script to
# run the experiments
python launching.py \
  --gpus "0,1" \
  --jobs-per-gpu 2 \
  --filters "ascent_forget,cifar10" \
  --output launch_ascent.sh
```

```
# Step 3: Run all jobs in parallel
# Can also be launched through sbatch
bash launch_ascent.sh
```

Results are saved automatically for KLoM scores and Margins. Triggering again a launch will skip the precomputed stages making the pipeline robust to interruptions and relaunches. The benchmark is also designed to be easily extensible. To add a new unlearning method, users simply define a new function in `unlearning.py` that follows the standard interface. The framework handles training, checkpointing, margin computation, and evaluation automatically. A typical implementation is fewer than 100 lines. Each method is registered in a central dispatch table and selected via a keyword in the YAML config. New hyperparameter sweeps can be added declaratively in `config.py`, and the resulting YAML files are generated with a single command. This modular design enables rapid and reproducible experimentation with minimal boilerplate.

4. Conclusions and Future Work

We collected a benchmark based on (GRP+24) that makes evaluating machine unlearning both practical and rigorous. By providing precomputed model ensembles, oracle outputs, and tested evaluation tools, our framework allows researchers to compare unlearning methods under standardized and reproducible conditions. The infrastructure is designed to be easy to use and supports reliable evaluation with strong baselines, such as retraining without the forget set. Our main evaluation metric, KLoM, offers a clear and tractable way to measure how close unlearned model predictions are to those of oracle models. We believe this benchmark lowers the barrier to entry and will help accelerate progress in developing more effective and efficient data unlearning algorithms along with better comparing of more efficient heuristic evals.

This benchmark can be extended in several promising directions. First, we are actively working on incorporating the Gaussian Unlearning Score (GUS) (PDL+24) unlearning evaluation, which introduces a complementary perspective by quantifying unlearning efficacy through data poisoning reversibility. We aim to integrate this metric into our framework after further validation and robustness testing. The code is implemented and undergoing integration.

In a similar line, we are interested in extending our evaluations to support LLMs more broadly. Improving upon other current methods evaluated on synthetic forget tasks, such as TOFU (MFS+24), and in understanding how unlearning techniques can scale to generative settings. We provided the initial discussion in Section 2.3.

We also consider the introduction of a public leaderboard, similar in spirit to RobustBench (CAS+20), to facilitate community engagement and transparent reporting of results. A key goal here is to also allow external researchers to contribute their own margin data, encouraging a collaborative and open environment.

Finally, we want to improve dataset and model coverage. Although CIFAR-10 and Living-17 provide useful testbeds, community input should guide the addition of new datasets and model sizes, especially for settings considering generative AI applications. We welcome feedback and contributions from the community to help determine which of these directions would be most impactful in practice.

References

[CAS+20] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[CCN+22] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[GRP+24] Kristian Georgiev, Roy Rinberg, Sung Min Park, Shivam Garg, Andrew Ilyas, Aleksander Madry, and Seth Neel. Attribute-to-delete: Machine unlearning via datamodel matching, 2024.

[HST+24] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy, 2024.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, 2009.

[LPG+24] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Liu, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

[MFS+24] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

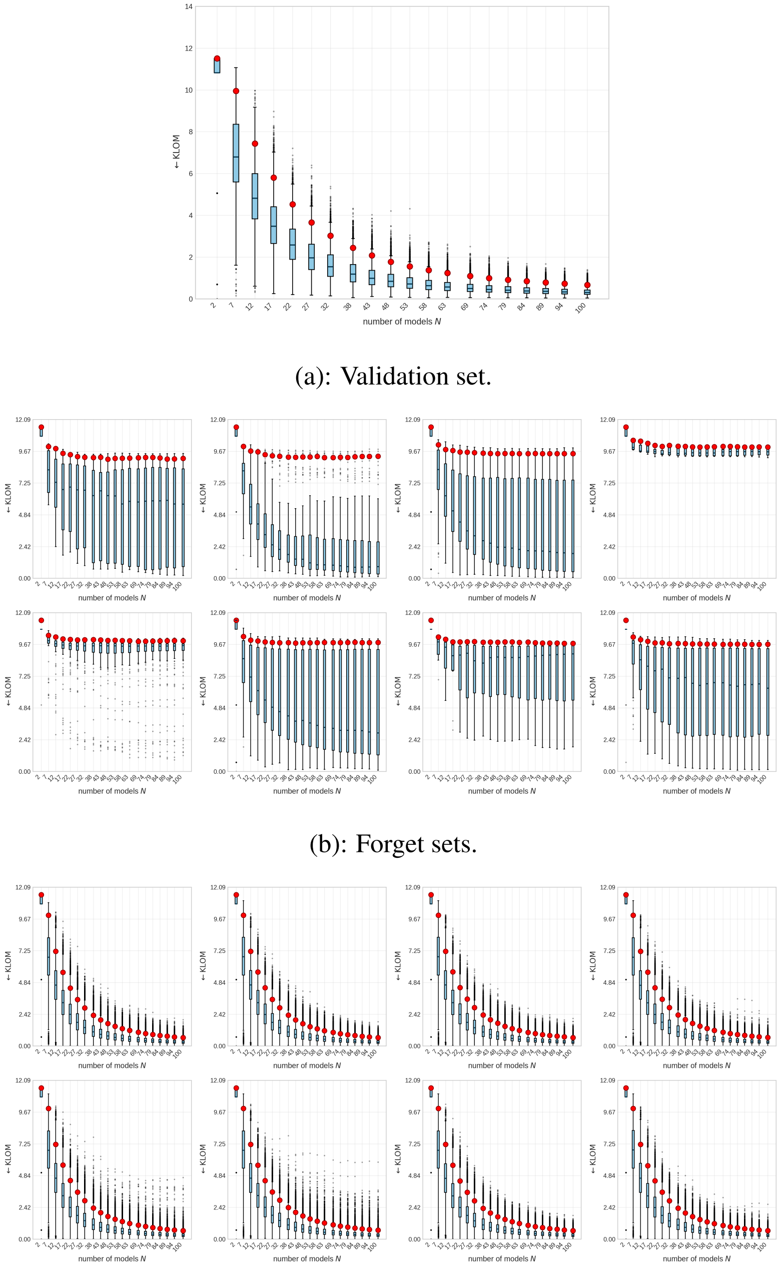
[PDL+24] Martin Pawelczyk, Jimmy Z Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine unlearning fails to remove data poisoning attacks. *arXiv preprint arXiv:2406.17216*, 2024.

[QPL+24] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep, 2024.

[STM21] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations (ICLR)*, 2021.

A. Appendix

A.1. KLoM sensitivity to number of compared models



(a): Validation set.

(b): Forget sets.

(c): Retain sets.

Figure 2: KLoM scores between pre-trained and oracle models scores on as a function of the number of compared models N on CIFAR10 sets (1-8) (GRP+24). The figure presents results for three data categories: (a) **Validation set**: a held-out test dataset, consistent across all forget configurations. (b) **Forget sets**: distributions for data points intended for unlearning. (c) **Retain sets**: distributions for data points to be preserved post-unlearning. In all panels, boxplots illustrate the KLoM value distributions for N ranging from 2 to 100. The red marker (●) represents the 95-th percentile of KLoM scores. Lower KLoM values indicate better alignment of the pre-trained models with the oracle models and are expected in the Retain and Validation sets. We find $N = 100$ to be sufficient for a reliable comparison.