Characterizing datapoints via Second-Split Forgetting

Pratyush Maini [1] **Saurabh Garg** [1] **Zachary C. Lipton** [1] **J. Zico Kolter** [1 2]

Abstract

The dynamics by which neural networks learn and forget examples throughout training has emerged as an object of interest along several threads of research. In particular, researchers have proposed metrics of example *hardness* based on these dynamics, including (i) the epoch at which examples are first correctly classified; (ii) the number of times their predictions flip during training; and (iii) whether their prediction flips if they are held out. However, an example might be considered *hard* for several distinct reasons, such as being a member of a rare subpopulation, being mislabeled, or being fundamentally ambiguous in their class. In this paper, we focus on the *second-split forgetting time* (SSFT): the epoch (if any) after which an original training example is forgotten as the network is fine-tuned on a randomly held out partition of the data. Across multiple benchmark datasets and modalities, we demonstrate that *mislabeled* examples are forgotten quickly, and seemingly *rare* examples are forgotten comparatively slowly. By contrast, metrics only considering the first split learning dynamics struggle to differentiate the two. Additionally, the SSFT tends to be robust to the choice of architecture, optimizer, and random seed. From a practical standpoint, the SSFT (i) can help to identify mislabeled samples, the removal of which improves generalization; and (ii) can provide insights about failure modes.

1. Introduction

While deep neural networks generalize remarkably well on unseen data (Krizhevsky et al., 2012), they are also known to be capable of memorizing noise (Zhang et al., 2021). Some efforts to reconcile the generalization power and expressivity of deep nets have turned towards learning dynamics, with researchers noting that neural networks tend to learn cleanly labeled examples before mislabeled examples (Liu et al., 2020), and more generally, exhibit a bias towards learning *simpler* patterns, for several intuitive notions of simplicity (Shah et al., 2020; Mangalam and Prabhu, 2019) Investigating this relationship between memorization and generalization, Feldman (2020) introduce a theoretical model to argue that memorization of *rare* examples (for some notion of rarity), maybe required to achieve close-to-optimal generalization. Broadly, works in this area tend to characterize examples as belonging either to *prototypical groups* or *memorized exceptions* (Feldman and Zhang, 2020; Jiang et al., 2020; Carlini et al., 2019). Adapting these intuitions to real datasets, Feldman (2020) propose rating the degree to which an example is memorized based on whether its predicted class flips when it is excluded from the training set. These, and other works (Hooker et al., 2019; Toneva et al., 2018) have proposed many metrics for characterizing example difficulty with Carlini et al. (2019) comparing five such metrics. However, while many of these works distinguish some notion of *easy* versus *hard* samples, they seldom (i) offer a finer resolution for distinguishing among different types of hard examples; (ii) reason about the observed separation between easy and hard samples. Existing metrics tend to group together examples that are difficult because they are a member of a rare subpopulation, mislabeled, fundamentally ambiguous in their class, or contain complex patterns.

In this paper, we propose to additionally consider a new metric, Second-Split Forgetting Time (SSFT), calculated based on the forgetting dynamics that arise as training examples are forgotten when a neural network continues to train on a second, randomly held out data partition. SSFT is defined as the fine-tuning epoch after which a first-split training example is no longer classified correctly. We find that SSFT identifies mislabeled examples remarkably well but does little to separate out under- versus over-represented subpopulations. Conversely, metrics based on the (first-split) training dynamics are more discriminative for separating these populations but less useful for detecting mislabeled examples. We leverage the complementarity of first- and second-split metrics, showing that by jointly visualizing the two, we can produce a richer characterization of the training examples.

In our experiments, we operationalize several notions of hard examples, namely: (i) **mislabeled** examples, for which

[*]Equal contribution [1]Carnegie Mellon University [2]Bosch Center for AI. Correspondence to: Pratyush Maini <pratyush.maini@cmu.edu>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

the original label has been flipped to a randomly chosen incorrect label; (ii) **rare** examples, which belong to underrepresented subpopulations; and (iii) **complex** examples, which belong to subpopulations for which the classification task is more difficult (details in Section 2.1). We perform specific ablation studies with datasets complicated by just one type of hard example (Section 3.1), and show how SSFT can help to distinguish among these categories of examples. We observe that during second-split training, neural networks (i) first forget mislabeled examples from the first split; (ii) only slowly begin to forget *rare* examples (e.g., from underrepresented sub-populations) unique to the first training set; and (iii) do not forget complex examples.

This separation of hard example types has multiple practical applications. **First**, we can use the method to identify noisy examples, the removal of which improves generalization: while the removal of hard examples according to first-split learning time degrades the performance of the classifier, the removal of hard examples according to SSFT can actually *improve* generalization. This is especially beneficial when e.g., training on synthetic data (produced by a generative model) or mislabeled data. **Second**, we show how SSFT can identify failure modes of machine learning models. For example, in a task of classifying horses and planes from the CIFAR-10 dataset, we find that training examples containing horses with sky backgrounds and planes with green backgrounds are among the earliest forgotten—indicating that the model relies on the background as a spurious feature.

2. Method

We aim to *characterize* the hardness of different datapoints in a dataset $\mathcal{S}_A = \{\mathbf{x}_i, \mathbf{y}_i\}^n$ such that $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}$. For this characterization, we augment each datapoint $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_A$ with parameters $(\mathbf{fslt}_i, \mathbf{ssft}_i)$ where $\mathbf{fslt}_i$ quantifies the first-split learning time (FSLT), and $\mathbf{ssft}_i$ quantifies the second-split forgetting time (SSFT) of the sample.

Procedure We train a model f on $\mathcal{S}$ to minimize the empirical risk: $\mathcal{L}(\mathcal{S}; f) = \sum_i \ell(f(\mathbf{x}_i), \mathbf{y}_i)$. We use f_A to denote a model f (initialized with random weights) trained on $\mathcal{S}_A$ until convergence (100% accuracy on $\mathcal{S}_A$). We then train a model initialized with f_A on a held-out split $\mathcal{S}_B \sim \mathcal{D}^n$ until convergence. We denote this model with $f_{A\to B}$. To obtain parameters $(\mathbf{fslt}_i, \mathbf{ssft}_i)$, we track per-example predictions $(\hat{\mathbf{y}}_i^t)$ at the end of every epoch (t^{th}) of training. We train the model with cross-entropy loss using SGD.

Definition 2.1 (First-Split Learning Time)**.** *For $\{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A$, learning time is defined as the earliest epoch during the training of a classifier f on $\mathcal{S}_A$ after which it is always classified correctly, i.e.,*

$$\mathbf{fslt}_i = \underset{t^*}{\text{argmin}}\ (\hat{\mathbf{y}}_{i,(A)}^t = \mathbf{y}_i \quad \forall t \geq t^*) \quad \forall \{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A.$$

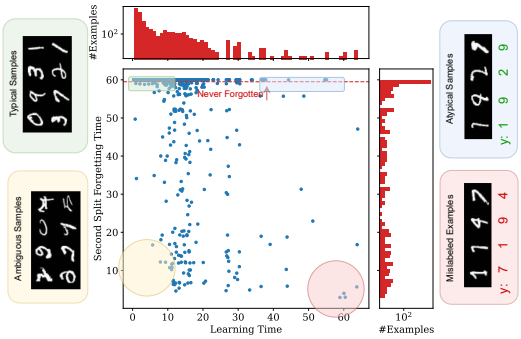


Figure 1: FSLT and SSFT for MNIST images.

Definition 2.2 (Second-Split Forgetting Time)**.** *Let $\hat{\mathbf{y}}_{i,(A\to B)}^t$ to denote the prediction of sample $\{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A$ after training $f_{(A\to B)}$ for t epochs on $\mathcal{S}_B$. Then, for $\{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A$ forgetting time is defined as the earliest epoch after which it is never classified correctly, i.e.,*

$$\mathbf{ssft}_i = \underset{t^*}{\text{argmin}}\ (\hat{\mathbf{y}}_{i,(A\to B)}^t \neq \mathbf{y}_i \quad \forall t \geq t^*) \quad \forall \{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A.$$

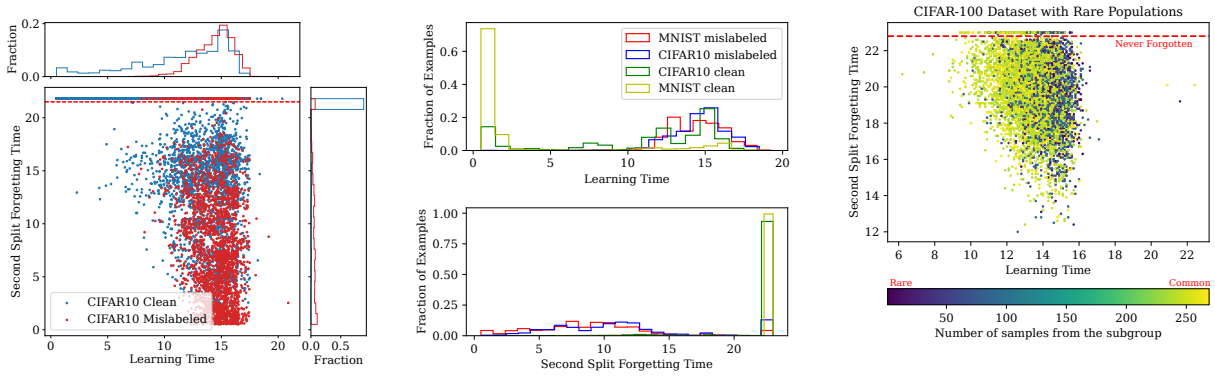
2.1. Example Characterization

We characterize example hardness via three sources of learning difficulty: **(i) Mislabeled Examples:** Those datapoints whose label has been flipped to an incorrect label uniformly at random. **(ii) Rare Examples:** Datapoints that belong to such sub-populations of the original distribution that have a low probability of occurrence. In particular, there exist $O(1)$ examples from such sub-populations in a given dataset. In Section 3.1 we describe how we operationalize this notion in the case of the CIFAR-100 dataset. **(iii) Complex Examples:** These constitute samples that are drawn from subgroups in the dataset that require either (1) a hypothesis class of high complexity; or (2) higher sample complexity to be learnt relative to examples from rest of the dataset. We leave the definition of complex samples mathematically imprecise, but with the same intuitive sense as in prior work (Shah et al., 2020). For instance, in a dataset composed of the union of MNIST and CIFAR-10 images, we would consider the subpopulation of CIFAR-10 images to be more *complex*.

3. Empirical Investigation

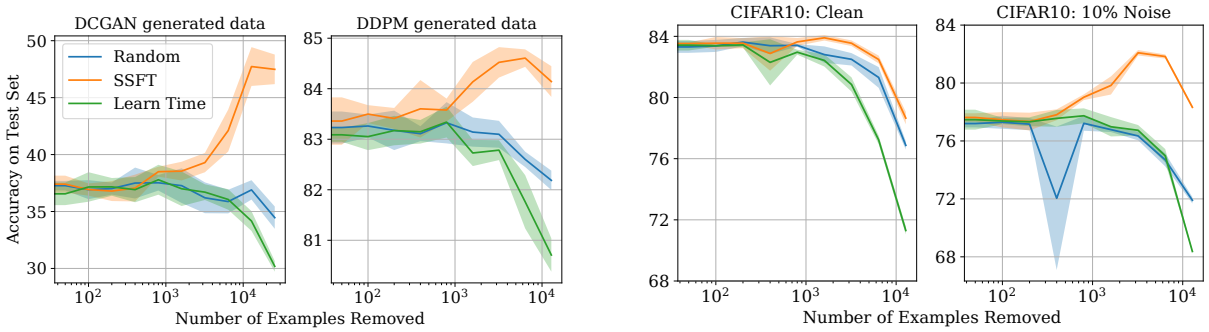
We describe complete training details in Appendix B.1.

Learning-Forgetting Spectrum for Image Domain In Figure 1, we show representative examples in the four quadrants of the learning-forgetting spectrum. More specifically, we find that the examples forgotten fastest and learned last are mislabeled. And the ones learned early and never forgotten once learned are characteristic simple examples of



(a) Mislabeled Examples (b) Complex Examples (c) Rare Examples

Figure 2: (a) Mislabeled samples may be learned as slowly as a high fraction of typical samples, but they are forgotten much faster. (b) FSLT distinguishes complex (CIFAR10 clean) and simple (MNIST clean) examples, but SSFT does not. On the contrary, FSLT can not distinguish clean and mislabeled examples of CIFAR10, while the SSFT can. (c) FSLT is able to distinguish examples based on the sub-group frequency, however, SSFT has a low correlation with the sub-group frequency.



(a) Synthetically generated CIFAR10-like data (b) CIFAR10 with and without label noise

Figure 3: Accuracy on CIFAR-10 test set after removing training examples (i) randomly, (ii) with the lowest SSFT, and (ii) with the highest FSLT. Removing examples based on SSFT helps improve the generalization on the original test set.

the MNIST dataset. Examples in the first and third quadrant are seemingly atypical and ambiguous respectively. Visualizations for other datasets can be found in Appendix B.2.

3.1. Ablation Experiments

We design specific experimental setups to capture the three notions of hardness as defined in Section 2.

Mislabeled Examples We sample 10% datapoints from both the first and second split of the CIFAR-10 dataset, and randomly change their label to an incorrect label. Figure 2a shows the learning-forgetting spectrum for the dataset. In the adjoining density histograms, note that a large fraction of the mislabeled and correctly labeled examples are learned at the same time. However, during second-split training, the mislabeled examples are forgotten quickly whereas a large fraction of the clean examples are never forgotten, allowing SSFT to succeed in distinguishing mislabeled samples.

Complex Examples We create a joint dataset that contains the union of both MNIST and CIFAR-10 examples. This is motivated by work in simplicity bias (Shah et al., 2020) that argues that neural networks learn simpler features first. We also add 10% label noise to each of the datasets to understand the learning and forgetting time relationship of a sample that is complex or mislabeled. In Figure 2b, we show the FSLT and SSFT for MNIST and CIFAR-10 samples. A high fraction of the CIFAR-10 (complex) samples learn at the same speed as the mislabeled samples. However, when looking at the SSFT, we are able to draw a strong separation between the mislabeled samples and complex samples. This indicates that the complex samples are learnt slowly, but have no strong tendency of being forgotten once learnt.

Rare Examples The CIFAR-100 (Krizhevsky et al., 2009)

dataset is a 100-class classification task. The dataset contains 20 superclasses, each containing 5 subclasses. We create a 20-class classification dataset with long tails simulated through the 5 sub-classes within each superclass. The number of examples in each subgroup for a given superclass is given by $\{500, 250, 125, 62, 31\}$ respectively (exponentially decaying). In order to remove any other effects of example hardness (either within a subgroup, or among subgroups), we randomize both the chosen subset of examples and the ordering of the majority and minority groups between each superclass, by training the model on 20 such random splits and aggregating learning and forgetting statistics over these runs. In Figure 2c, we show a scatter plot for the FSLT and SSFT, colored by the frequency of the group a particular example belongs to. We observe that FSLT strongly correlates with the size of the subgroup, whereas the SSFT has a very low correlation with the rareness of a sample.

Figure 4: We observe that the model quickly forgets planes with green backgrounds and horses on blue backgrounds.

3.2. Dataset Cleansing

Generative models are capable of mimicking the distribution of a given dataset. We generate synthetic datasets of CIFAR10-like samples using (i) DDPM (denoising diffusion model (Ho et al., 2020)); and (ii) DCGAN (Deep Convolutional GAN (Radford et al., 2015)). In both cases, we assign pseudo-labels using the BiT model (Kolesnikov et al., 2020) as in prior work (Nakkiran et al., 2021). We collect a sample of 50,000 training examples and record the generalization performance on CIFAR-10 as we remove 'hard' samples, as evaluated by various metrics. In Figure 3, we can see that removing the most easily forgotten examples can benefit by up to 10% generalization accuracy on the clean test set of CIFAR-10. In case of the synthetic data generated using DDPM, the gains in generalization performance are under 2%. We hypothesize that this is because the samples generated by DDPM are more representative of the typical distribution of CIFAR-10 than those generated by DCGAN.

3.3. Evaluating Example Utility

Recent works (Toneva et al., 2018; Feldman and Zhang, 2020) have argued for removing a large fraction of the less memorized examples, and keeping the memorized ones. We analyze the change in model generalization upon removing varying sizes of examples from the training set, as ranked by lowest SSFT and highest FSLT (Figure 3).

FSLT finds important samples As we remove more samples from the dataset based on the highest FSLT, the test set accuracy of the model is significantly lower than random guessing. This suggests that the utility of these samples is higher than random samples. Put in line with the hypothesis of memorization of rare example as proposed in (Feldman, 2020), we see that empirically, the examples that are slow to learn are important for the model's test set generalization.

SSFT removes pathological samples On the contrary, removing examples based on the SSFT helps improve model generalization (especially when there is label noise). Even in the setting when there is no label noise, in contrast to FSLT, we find that removing examples that were easily forgotten has a lower negative impact on the model's generalization as opposed to removing random samples. This suggests that the examples that are forgotten in the early epochs of second-split training hurt a model's generalization, and may not be characteristic samples of their particular class.

3.4. Characterizing Potential Failure Modes

Recent works have attempted to train classifiers on datasets that contain spurious features (Sagawa et al., 2019; Idrissi et al., 2021). However, a fundamental challenge is to first identify the spurious correlation that the classifier may be relying on. Only then can recent methods be trained to remove the reliance on spurious patterns. We train a ResNet-9 model to classify CIFAR-10 images of horses and airplanes. In Figure 4, we observe that the model forgets planes with green backgrounds and horses with blue backgrounds. This suggests that the model relied on the background as a spurious feature. By analyzing forgotten examples we can identify spurious features that a classifier associates with a class.

4. Conclusion

While many prior works investigate training time dynamics to characterize the hardness of examples, we enrich this literature with a complementary lens focused on the second-split forgetting time. We demonstrate the potential of SSFT to distinguish among rare, mislabeled, and complex examples; and also show the differences in the example properties captured by first-split and second-split metrics. Our work opens new lines of inquiry in future work that can utilize the separation of hard examples. First, we expect state of art methods in label noise identification to benefit by augmenting our approach. Further, we believe our ablations showing that complex, noisy, and mislabeled samples may all be learned slowly inspire future work that can unite different takes on the memorization-generalization research—early learning, simplicity bias, and singleton memorization.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Devansh Arpit, Stanislaw Jastrzbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

Woonhyuk Baek. Torchskeleton. `https://github.com/wbaek/torchskeleton`, 2019.

Robert John Nicholas Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=WWRBHhH158K`.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 123–132, 2021.

Nicholas Carlini, Úlfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. *ArXiv*, abs/1910.13427, 2019.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. *arXiv preprint arXiv:2002.10657*, 2020.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019.

Chen Cheng, John Duchi, and Rohith Kuditipudi. Memorize to generalize: on the necessity of interpolation in high dimensional linear regression. *arXiv preprint arXiv:2202.09889*, 2022.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 2020.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3326–3334, 2019.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. *arXiv preprint arXiv:2110.14503*, 2021.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. Characterizing structural regularities of labeled data in overparameterized models. *arXiv preprint arXiv:2002.03206*, 2020.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 967–972. IEEE, 2016.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgExaVtwr.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

Karttikeya Mangalam and Vinay Uday Prabhu. Do deep neural networks learn shallow learnable examples first? 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021 (12):124003, 2021.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021b.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

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. *arXiv preprint arXiv:1812.05159*, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Appendix
Characterizing Datapoints via Second-Split Forgetting

A. Related Work

Example Hardness. Several recent works quantify example hardness with various training-time metrics. Many of these metrics are based on first-split learning dynamics (Chatterjee, 2020; Jiang et al., 2020; Mangalam and Prabhu, 2019; Shah et al., 2020). Other works have resorted to properties of deep networks such as compression ability (Hooker et al., 2019) and prediction depth (Baldock et al., 2021). Carlini et al. (2019) study metrics centered around model training such as confidence, ensemble agreement, adversarial robustness, holdout retraining, and accuracy under privacy-preserving training. Closest in spirit to the SSFT studied in our paper are efforts in (Carlini et al., 2019; Toneva et al., 2018). Crucially, Carlini et al. (2019) study the KL divergence of the prediction vector after fine-tuning on a held-out set at a low learning rate, and do not draw any direct inference of the separation offered by their metric. Focusing on (first-split) forgetting dynamics, Toneva et al. (2018) defined a metric based on the number of forgetting events during training and identified sets of *unforgettable* examples that are never misclassified once learned. In our work, we find complementary benefits of analysis based on first- and second-split dynamics.

Memorization of Data Points. In order to capture the memorization ability of deep networks, their ability to memorize noise (or randomly labeled samples) has been studied in recent work (Zhang et al., 2021; Arpit et al., 2017). As opposed to the memorization of rare examples, the memorization of noisy samples hurts generalization and makes the classifier boundary more complex (Feldman, 2020). On the contrary, a recent line of works has argued how memorization of (atypical) data points is important for achieving optimal generalization performance when data is sampled from long-tailed distributions (Feldman, 2020; Brown et al., 2021; Cheng et al., 2022).

Simplicity Bias. Another line of work argues that neural networks have a bias toward learning simple features (Shah et al., 2020), and often do not learn complex features even when the complex feature is more predictive of the true label than the simple features. This suggests that models end up memorizing (through noise) the few samples in the dataset that contain the complex feature alone, and utilize the simple feature for correctly predicting the other training examples (Li et al., 2019; Allen-Zhu and Li, 2020).

Label Noise. Large-scale machine learning datasets are typically labeled with the help of human labelers (Deng et al., 2009) to facilitate supervised learning. It has been shown that a significant fraction of these labels are erroneous in common machine learning datasets (Northcutt et al., 2021b). Learning under noisy labels is a long-studied problem (Angluin and Laird, 1988; Natarajan et al., 2013; Jindal et al., 2016; Li et al., 2020). Various recent methods have also attempted to identify label noise (Northcutt et al., 2021a; Chen et al., 2019; Pleiss et al., 2020; Huang et al., 2019). While the focus of our work is not to propose a new method in this long line of work, we show that the view of forgetting time naturally distills out examples with noisy labels. Future work may benefit by augmenting our metric with SOTA methods for label noise identification.

B. Experimental Results

B.1. Experimental Setup

Architectures We perform experiments using four different model architectures—LeNet, ResNet-9 (Baek, 2019), ResNet-50, and Bert-base-cased (Devlin et al., 2018). Comparisons with model architectures are used in analysis of stability of the SSFT metric. For other numbers reported in tables and plots, we use the ResNet-9 model, unless otherwise stated.

Optimizer We experiment with three different learning rate scheduling strategies—cyclic learning rate schedule, cosine learning rate, and step decay learning rate. We test for two values of peak learning rate—0.1 and 0.01. All the model are trained using the SGD optimizer with weight decay 5e-4 and momentum 0.9, apart from the comparison with optimizers in Appendix C.1 where we also experiment with the Adam optimizer.

Training Procedure We train for a maximum of 100 epochs or until we have 5 epochs of 100% training accuracy. We first train on $\mathcal{S}_A$, and then using the pre-initialized weights from stage 1, train on $\mathcal{S}_B$ with the same learning parameters. All experiments can be performed on a single RTX2080 Ti. Code for running our experiments can be found here.

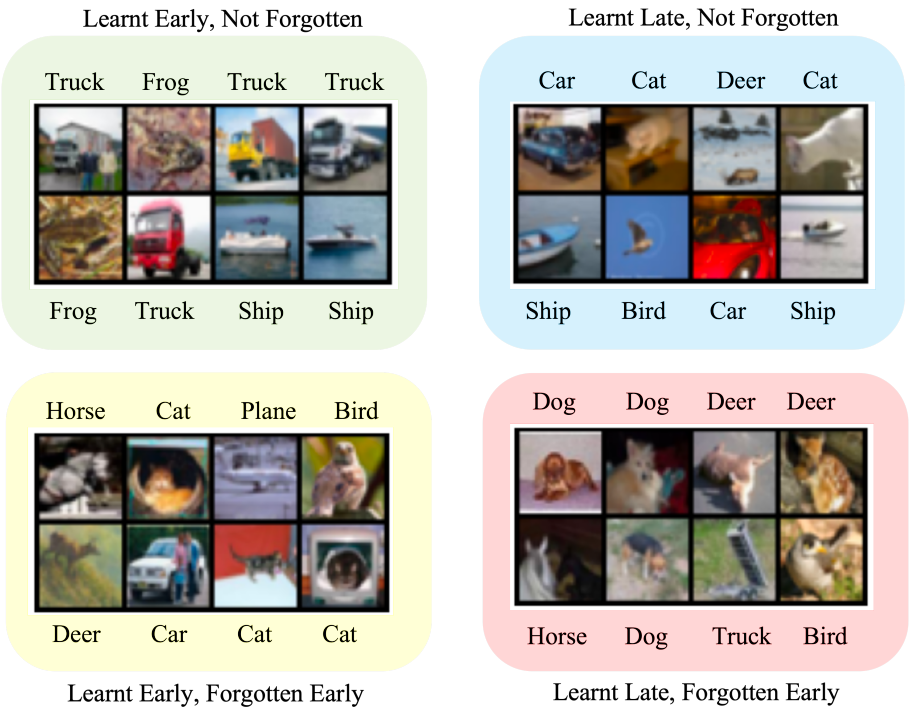


Figure 5: Examples from the CIFAR-10 dataset grouped based on their learning and forgetting time.

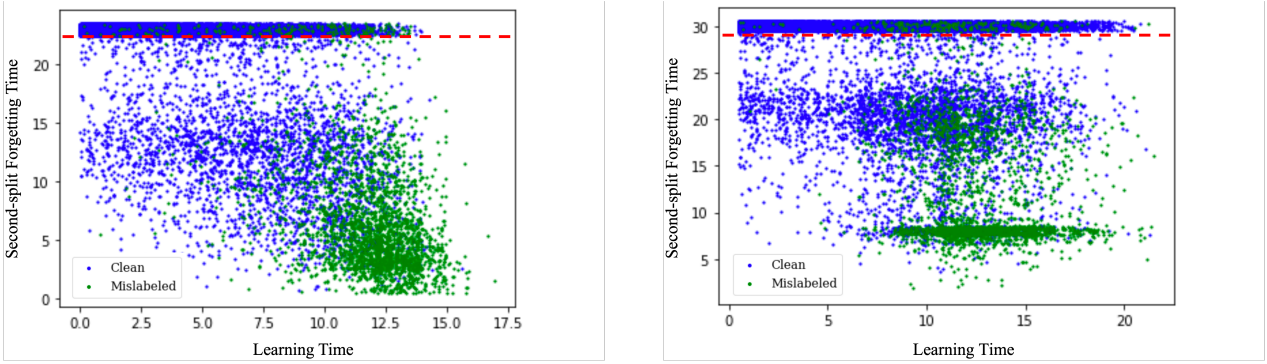
B.2. Image Datasets

In the main paper we present visualizations of training examples from the MNIST dataset based on which quadrant they lie on in the learning-forgetting graph. Here, we complement our findings by showing visualizations for the CIFAR-10 dataset. We note that CIFAR-10 dataset provide many different types of visibility patterns within the same class. Hence, examples may be learnt late due to belonging to a rare visibility pattern. In Figure 5, we see that the examples that were learnt earliest and never forgotten have similar visibility patterns—for instance all the trucks have a similar perspective. On the contrary, as we move to the first quadrant with examples that were learnt late but never forgotten, we see that all the examples are true to their semantic class, but these visibility patterns occur rarely. Finally, we also analyze the visualizations based on examples that were forgotten during the course of second-split training. While in the case of MNIST dataset, SSFT was able to remove the mislabeled examples well, we see that CIFAR-10 offers more challenges because examples may be ambiguous because of other reasons and may be forgotten owing to the model using spurious features.

C. Ablation Studies

We detail the experimental setup used to conduct our ablation studies directed towards understanding the learning and forgetting dynamics of rare and complex examples respectively.

Rare Examples The experiments to show the rate of learning for rare examples are inspired by the singleton hypothesis as proposed by Feldman (2020). The hypothesis was inspired by a long-tailed distribution of visibility patterns in the person and bus category of the PASCAL dataset. For example, the dataset contains many buses with the front visible, but very few buses that were captured from the rear or the side, and even fewer buses whose view is occluded by the presence of other objects infront of them. (Refer to Figure 1 in their work for more details.) In our work, we first attempted to leverage the same training set-up with the provided visibility patters. However, we noted that there wasn't a strong correlation between the frequency of an example's visibility pattern, and the rate at which it was learnt. We hypothesize that this is because there are other factors of example hardness that may make an example be learnt slowly or fast (such as complexity, as detailed in the next paragraph). This can lead to an example being learnt fast if it has a simple pattern yet occurs rarely. Especially when there are only $O(1)$ samples from a given sub-group (based on the visibility pattern), we can not make any claims



(a) (b)

Figure 6: FSLT (First-split learning time) is able to provide some degree of separation between mislabeled and clean samples when trained with the SGD optimizer (left), but fails when the model is trained using Adam (right) on the CIFAR-10 dataset.

based on singleton correlation alone.

Hence, in order to distill the frequency of occurrence of an example with other confounders that may influence its training-time, we created a long-tailed dataset from the CIFAR-100 dataset. CIFAR-100 is a dataset of 100 object classes, which can be further grouped into 20 super-classes. For instance, examples from categories *maple, oak, palm, pine, willow* all belong to the 'superclass' of *trees*. Similar division of 5 sub-classes is provided in the datasets for each of the superclasses. Each class contains 500 training examples and the overall dataset has 50,000 training examples.

As a first step towards creating a long-tailed dataset, we assign a fixed frequency ordering within the subgroups of a superclass. The most frequent subgroup has 500 examples in the training set, for the next most frequent subgroup, we randomly select 250 examples from the training set, and so on until the last sub group with 31 examples in the training set. This means that there are exactly 20 sub-groups in the final dataset with $\{500,250,125,62,31\}$ examples respectively. Irrespective of the class number, the task is to predict the corresponding superclass, that is, we reduce the problem to a 20-class classification problem. However, we track the learning and forgetting dynamics of examples from each of the 100 sub-groups separately, based on their group frequency. To remove any other confounders of example hardness, we (i) randomize the group frequency ordering of the sub-groups within a superclass (in case some classes are harder to learn than the others); and (ii) randomize the examples that were selected based on the group size (in case some examples were ambiguous or hard). We further split the dataset into two IID partitions to analyze the learning time and SSFT, and average the results over 20 random runs of the experiment. Experimental results are detailed in the main paper.

Complex Examples Prior works advocating for, and understanding the simplicity bias (Shah et al., 2020) have operationalized the notion of simplicity via the complexity of hypothesis class required to learn the distribution that a complex example may be sampled from. In particular, Shah et al. (2020) construct a synthetic dataset with MNIST and CIFAR-10 images vertically stacked on top of each other—with the part with MNIST images corresponding to the part of the combined image with *simpler* features, and the part with CIFAR-10 images corresponding to the part with *complex* features. They show that the model almost completely relies on the part of the image containing the MNIST digit even when it is less predictive of the true label. Inspired by this argument about the simplicity of features, we create a dataset that has the the union of images from the MNIST and the CIFAR-10 dataset. More specifically, we select classes from the MNIST dataset corresponding to digits $\{0, 1, 2, 3\}$, and classes from the CIFAR-10 dataset corresponding to {horses, airplanes, dog, frog} and label them from $\{0, 1, 2, 3\}$. This means that the model associates the label 0 to both the digit 0 and airplane class. The attempt of this experiment is to draw the link between the simplicity bias and the rate of learning. Experimental results are provided in the main paper.

C.1. Stability of our metric

Stability across architectures The forgetting of examples is a property of both the dataset and the model architecture. As a result, we find that just like the learning time, the forgetting time has a lower correlation between architectures. The average pearson correlation between the ResNet-9 and ResNet-50 models is 0.62 in case of the CIFAR10 dataset. However,

we note that the most forgotten examples generalize across datasets. That is, the average pearson correlation between the bottom 10% examples of the dataset is 0.87. This highlights how the forgetting metric is good for finding misaligned examples in the dataset, since they are not a property of the model architecture. We suspect that among the examples that are infrequently forgotten, the model campacity and other inductive biases of the model architecture may have a role in driving the average pearson correlation low.

Stability across optimizers Jiang et al. (2020) showed that changing the learning optimizer from SGD to Adam can lead to a significant change in the learning rate of examples from different levels of hardness (based on their regularity metric). More specifically, they find that examples with a low consistency score (closely correlated with learning speed) also get learnt fast when using the Adam optimizer. This suggests that using an optimizer like Adam at training time may have an impact on the ability of learning time based metrics to separate examples. In Figure 6, we contrast the ability of forgetting and learning time based metrics for identifying label noise when using the SGD and Adam optimizers. When using an optimizer such as SGD, the mislabeled samples are learnt slower than a large fraction of the training examples, and the learning time metric offers some degree of separation between the clean and mislabeled examples. However, when we use the Adam optimizer, it results in joint learning of a large fraction of both mislabeled and clean samples. Hence, offering a very low degree of separation. However, under the same training procedure, the SSFT still allows us to distinguish between the mislabeled and clean samples.

Stability across seeds and learning rates The pearson correlation for stability across seeds for the forgetting time metric is 0.83. This is higher than the corresponding learning time based metric (correlation 0.56). However, one of the drawbacks of our proposed metric is that the SSFT requires the use of an appropriate learning rate that allows the examples to be forgotten slowly. We provide more information about the same in the main paper.