Characterizing Datapoints via Second-Split Forgetting

Anonymous Author(s)
Affiliation
Address
email

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. Through theoretical analysis addressing overparameterized linear models, we provide insights into how the observed phenomena may arise.

1 Introduction

While deep neural networks generalize remarkably well on unseen data [28], they are also known to be capable of memorizing noise [46]. 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 [16, 17, 31], and more generally, exhibit a bias towards learning *simpler* patterns, for several intuitive notions of simplicity [18, 33, 41]. Investigating this relationship between memorization and generalization, Feldman [14] 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* [7, 15, 24]. Adapting these intuitions to real datasets, Feldman [14] 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 [8, 20, 33, 41, 45] have proposed many metrics for characterizing example difficulty with Carlini et al. [7] comparing five such metrics. However, while many of these works distinguish some notion of *easy* versus *hard* samples, they seldom offer (i) a finer resolution for distinguishing among different types of hard examples; (ii) progress towards explaining how the observed separation between easy and hard samples arises. Existing metrics tend to group together examples that are difficult because they are a member of a rare subpopulation, are mislabeled, fundamentally ambiguous in their class, or contain complex patterns.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

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 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 3.2). We perform specific ablation studies with datasets complicated by just one type of hard example (Section 4.3), 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 subpopulations) 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 labels: On CIFAR-10 with 10% added class noise, SSFT achieves 0.94 AUC for identifying mislabeled samples, while the first-split metrics range in AUC between 0.58 to 0.90. **Second**, the method can also help improve generalization in noisy data settings: 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. **Third**, we show how SSFT can identify failure modes of machine learning models. For example, in a simplified task classifying between horses and airplanes in the CIFAR-10 dataset, we find that training examples containing horses with sky backgrounds and airplanes with green backgrounds are among the earliest forgotten—indicating that the model relies on the background as a spurious feature. **Last**, we also add that our metric is robust across multiple seeds, and the earliest forgotten examples are robust across architectures. Across multiple optimizers, SSFT distinguishes mislabeled samples, whereas first-split metrics appear more sensitive to the choice of optimizer.

Finally, we investigate second-split dynamics theoretically, analyzing overparametrized linear models [44]. We introduce notions of mislabeled, rare, and complex examples appropriate to this toy model. Our analysis shows that mislabeled examples from the first split are forgotten quickly during second-split training whereas rare examples are not. However, as we train for a long time, rare examples from the first split are eventually forgotten as the model converges to the minimum norm solution on the second split while predictions on complex examples remain accurate with high probability.

2 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 [8, 24, 33, 41]. Other works have resorted to properties of deep networks such as compression ability [20] and prediction depth [5]. Carlini et al. [7] 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 [7, 45]. Crucially, Carlini et al. [7] 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. [45] 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 [3, 46]. As opposed to the memorization of rare examples, the memorization of noisy samples hurts generalization and makes the classifier boundary more complex [14]. 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 [6, 10, 14].

Simplicity Bias. Another line of work argues that neural networks have a bias toward learning simple features [41], 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 [1, 30].

Label Noise. Large-scale machine learning datasets are typically labeled with the help of human labelers [11] to facilitate supervised learning. It has been shown that a significant fraction of these labels are erroneous in common machine learning datasets [37]. Learning under noisy labels is a long-studied problem [2, 25, 29, 35]. Various recent methods have also attempted to identify label noise [9, 22, 36, 38]. 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.

3 Method

The primary goal of our work is to *characterize* the hardness of different datapoints in a given dataset. Suppose we have a dataset $\mathcal{S}_A = \{\mathbf{x}_i, \mathbf{y}_i\}^n$ such that $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}$. For the purpose of 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. To obtain these parameters, we next describe our proposed procedure.

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. Unless specified otherwise, we train the model with cross-entropy loss using Stochastic Gradient Descent (SGD).

Definition 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.,*

$$\mathbf{fslt}_i = \underset{t^*}{\operatorname{argmin}} \; (\hat{\mathbf{y}}_{i,(A)}^t = \mathbf{y}_i \;\; \forall t \geq t^*) \quad \forall \{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A. \tag{1}$$

Definition 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.,*

$$\mathbf{ssft}_i = \underset{t^*}{\operatorname{argmin}} \; (\hat{\mathbf{y}}_{i,(A \to B)}^t \neq \mathbf{y}_i \quad \forall t \geq t^*) \quad \forall \{\mathbf{x}_i, \mathbf{y}_i\} \in \mathcal{S}_A. \tag{2}$$

3.1 Baseline Methods

We provide a brief description of metrics for example hardness considered in recent comparisons [24].

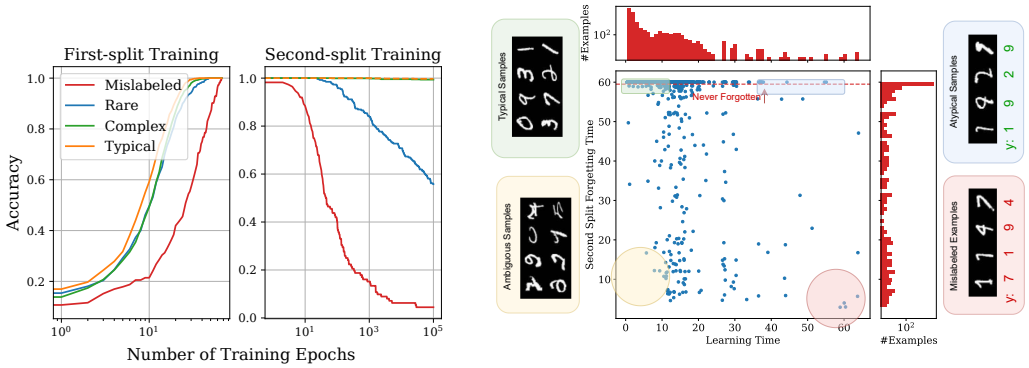
Number of Forgetting Events: (n_f). An example $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}$ undergoes a forgetting event when the accuracy on the example decreases between two consecutive updates. Toneva et al. [45] analyzed the total number of such events n_f during the training of a neural network to identify hard examples.

Cumulative Learning Accuracy: (acc_l). Jiang et al. [24] suggest that rather than using the learning time (Definition 1), using the number of epochs during training when a machine learning model correctly classifies a given sample is a more stable metric for predicting example hardness.

Cumulative Learning Confidence: (conf_l). Similar to acc_l, conf_l measures the cumulative softmax confidence of the model towards the correct class over the course of training.

3.2 Example Characterization

We characterize example hardness via three sources of learning difficulty: **(i) Mislabeled Examples:** We refer to mislabeled examples as those datapoints whose label has been flipped to an incorrect



(a) (b)

Figure 1: Rate of Learning and Forgetting of examples for different groups in the synthetic dataset. While first-split training is not able to distinguish between rare and complex examples, second-split training succeeds in distinguishing them. Additionally, second-split training separates mislabeled examples from the rest relatively better than first-split training. (b) Visualization of first-split learning and second-split forgetting times when training LeNet model on the MNIST dataset.

label uniformly at random. **(ii) Rare Examples:** We assume that rare examples belong to 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 4.3 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 sub-groups 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 [3, 41]. 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*.

4 Empirical Investigation of First- and Second-Split Forgetting Dynamics

4.1 Experimental Setup

Datasets We show results on a variety of image classification datasets—MNIST [12], CIFAR-10 [27], and Imagenette [21]. For experiments in the language domain, we use the SST-2 dataset [43]. For each of the datasets, we split the training set into two equal partitions $(\mathcal{S}_A, \mathcal{S}_B)$. For experiments with mislabeled examples, we simulate mislabeled examples by randomly selecting a subset of 10% examples from both the partitions and changing their label to an incorrect class.

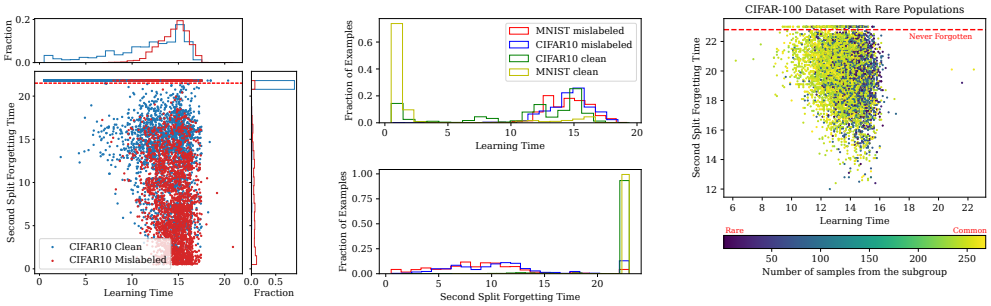
Training Details Unless otherwise specified, we train a ResNet-9 model [4] using SGD optimizer with weight decay 5e-4 and momentum 0.9. We use the cyclic learning rate schedule [42] with a peak learning rate of 0.1 at the 10th epoch. We train for a maximum of 100 epochs or until we have 5 epochs of 100% 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. Complete hyperparameter details are available in Appendix B.1.

4.2 Learning-Forgetting Spectrum for various datasets

Synthetic Dataset We consider data $(\mathbf{x}, \mathbf{y})$ sampled from a mixture of multiple distributions $\mathcal{D}_g$, s.t. $\mathbf{x} \in \mathbb{R}^d$. $\mathcal{D}_g$ denotes the gth group and has a sampling frequency of π_g. Each group $\mathcal{D}_g \equiv (\mathcal{X}_g, \{\mathbf{y}_g\})$, i.e., the true label for all the samples drawn from a given group is the same, and the examples in each group are non-overlapping. Each group is parametrized by a set of $k \ll d$ unique indices $\mathcal{I}_g \subset [d]$ such that $\mathcal{I}_i \cap \mathcal{I}_j = \phi$ for $i \neq j$. The discriminative characteristic of each group is the vector $\mathbf{u}_g$,

Sentences in SST-2 dataset with smallest forgetting time	Label
The director explores all three sides of his story with a sensitivity and an inquisitiveness reminiscent of Truffaut	Neg
Beneath the film's obvious determination to shock at any cost lies considerable skill and determination , backed by sheer nerve	Neg
This is a fragmented film, once a good idea that was followed by the bad idea to turn it into a movie	Pos
The holiday message of the 37-minute Santa vs. the Snowman leaves a lot to be desired.	Pos
Epps has neither the charisma nor the natural affability that has made Tucker a star	Pos
The bottom line is the piece works brilliantly	Neg
Alternative medicine obviously has its merits ... but Ayurveda does the field no favors	Pos
What could have easily become a cold, calculated exercise in postmodern pastiche winds up a powerful and deeply moving example of melodramatic moviemaking	Neg
Lacks depth	Pos
Certain to be distasteful to children and adults alike , Eight Crazy Nights is a total misfire	Pos

Table 1: First-split sentences that were forgotten by the 3rd epoch of second-split training of a BERT-base model on the SST-2 dataset. Notice that all of these forgotten examples are mislabeled.



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

Figure 2: Ablation experiments to distinguish the learning and forgetting dynamics for specific types of hard examples. (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.

such that, $[\mathbf{u}_g]_i = 1$ if $i \in \mathcal{I}_g$ else $0 \; \forall i \in [d]$. Then for any sample $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$:

$$P_{distX_g}(\mathbf{x}) = \pi_g; \quad \mathbf{x}|\mathcal{X}_g \sim \mathcal{N}(0, \sigma^2 I_n) + \mu_g \mathbf{u}_g. \tag{3}$$

For our simulation, we consider a 10 class-classification problem, with $\mu_g = 5$ for typical groups, and $\mu_g = 4$ for complex groups (higher signal to noise ratio). For any sample drawn from a rare group, we have $O(1)$ samples from that group in the entire dataset ($\mathcal{S}_A \cup \mathcal{S}_B$). Mislabeled samples are only generated from the majority typical groups. In Figure 1a, we show the rate of learning and forgetting of examples from each of these categories. We note that in the second-split training, the mislabeled examples are quickly forgotten, and the complex examples are never forgotten. The rare examples are forgotten slowly. In Section 5 we will theoretically justify the observations in the synthetic dataset and show that the rare examples are expected to be forgotten as we train for an infinite time.

Image Domain In Figure 1b, 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 the MNIST dataset. Examples in the first and third quadrant are seemingly atypical and ambiguous respectively. Similar visualizations for other image datasets can be found in Appendix B.2.

Other Modalities The forgetting and learning dynamics occur broadly across modalities apart from images. We repeat the same problem setup on the SST-2 [43] dataset for sentiment classification. We fine-tune a pre-trained BERT-base model [13] successively on two disjoint splits of the dataset. In Table 1, we provide a list of the earliest forgotten samples when we train a BERT model on the second split of SST-2 dataset. The results suggest that SSFT is able to identify mislabeled samples.

Method →	**fslt**	$\mathbf{acc}_l$	**ssft** (Ours)	$\mathbf{acc}_f$ (Ours)	$\mathbf{conf}_l$	$\mathbf{n}_f$	Joint (Ours)
Imagenette	0.834	0.912	0.931	0.941	0.786	0.781	0.957
CIFAR10	0.74	0.90	0.938	0.941	0.947	0.580	0.958
MNIST	0.973	0.998	0.997	0.998	0.965	0.377	0.998

Table 2: AUC for identification of label noise using various metrics for example hardness across different datasets. Across all datasets, our **ssft** metric outperforms alternative baselines. We introduce $\mathbf{acc}_f$ as the cumulative accuracy on the second-split training, inspired by previous work that suggests using cumulative accuracies helps make first-split learning time more stable [24]. All other notations are described in Section 3. In the case of the Joint method, we select new prediction ranks based on the combined learning and forgetting ranks, further improving over the **ssft** metric alone.

4.3 Ablation Experiments

We design specific experimental setups to capture the three notions of hardness as defined in Section 3.

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.

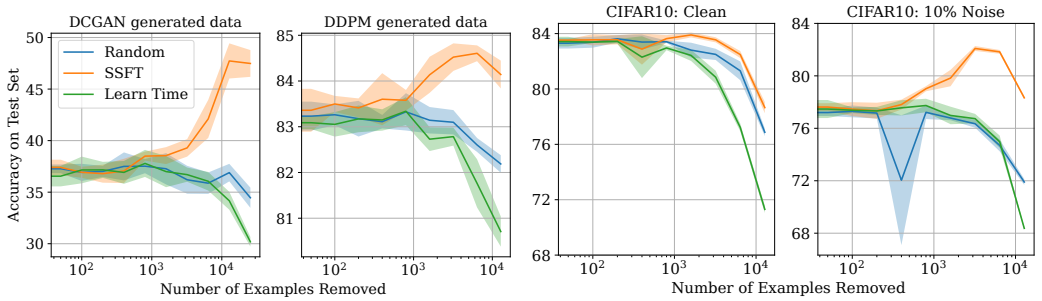
Complex Examples We generate a joint dataset that contains the union of both MNIST and CIFAR-10 examples. This is motivated by work in simplicity bias [41] that argue that neural networks learn simpler features first. We also add 10% labeled noise to each of the datasets in the union to understand the learning and forgetting time relationship of a sample that is complex or mislabeled together. In Figure 2b, we show the FSLT and SSFT for MNIST and CIFAR-10 samples. We note that 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 complexity of a sample has low correlation with its tendency to be forgotten once learnt, but a high correlation with being learned slowly.

Rare Examples The CIFAR-100 [27] 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. More specifically, the number of examples in each subgroup for a given superclass is given by {500, 250, 125, 64, 32} respectively (exponentially decaying with a factor of 2). This is done to simulate the hypothesis of dataset subgroups following a Zipf distribution [47] as argued for by Feldman [14]. This dataset is further divided into two equal splits to analyze the learning-forgetting dynamics. 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.

We provide further ablations to show that FSLT is able to identify hard and rare examples, but SSFT shows nearly no discriminative power at finding the two in Appendix C.

4.4 Dataset Cleansing

Identifying Label Noise We present AUC scores for detection of label noise via various popular methods in example difficulty literature, across various datasets in Table 2. We note that (i) cumulative predictions over the course of training help stabilize both the learning time and forgetting time metrics; (ii) for simple datasets such as MNIST with few ambiguous images, all of the baseline methods have very high AUC (greater than 0.99) in finding noisy inputs. However, in datasets such as CIFAR-10 and Imagenette, we find that second-split forgetting metrics do better than first-split training metrics. Finally, we also compare the use of both forgetting and learning time to find noisy samples, and we find a small improvement in the results of just using the forgetting time. While we do not make explicit



(a) Synthetically generated CIFAR10-like data

(b) CIFAR10 with and without label noise

Figure 3: Accuracy on CIFAR-10 test set after removing a varying number of samples from the training set based on (i) random choice, (ii) examples with the lowest SSFT, and (ii) the highest FSLT. Removing examples based on SSFT helps improve the generalization on the original test set.

comparisons with other state of art methods dedicated to finding label noise, our results suggest that augmenting second split forgetting time information may help improve their results. As also observed in recent work [24], we find that the number of forgetting events ($\mathbf{n}_f$) [45] is an unreliable indicator of mislabeled samples. We hypothesize that this is because of the fact that mislabeled examples may often be (first) learnt very late, hence their count of total forgetting events is also low.

Cleaning synthetically generated datasets 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 [19]); and (ii) DCGAN (Deep Convolutional GAN [39]). In both cases, we assign pseudo-labels using the BiT model [26] as in prior work [34]. 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.

Note: The ability to train on a second split allows SSFT the *unique* opportunity to train on a clean split of CIFAR-10 in order to assess the alignment of the synthetic samples with the oracle samples. As a result, the SSFT is much more effective in filtering out ambiguous first-split synthetic examples.

4.5 Evaluating Example Utility

Recent works [15, 45] have argued for removing a large fraction of the less memorized examples, and keeping the memorized ones. We will 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). In the presence of noisy examples, removing samples based on the SSFT helps improve generalization, whereas FSLT does not do much better than random. We draw the following inferences:

FSLT finds important samples As we remove more samples from the dataset, the accuracy of the model trained after samples are removed based on the highest FSLT 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 [14], 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.

Practitioner's view From the AUC numbers in Table 2, it may appear that removing examples via learning-based metrics such as learning time and cumulative learning accuracy also provides a high

rate of removal of noisy samples. However, when we observe the example utility graphs in Figure 3, we draw the inference that the examples that are learned late, are often important examples (such as rare memorized examples). However, even when SSFT fails to capture the correct noisy examples, it still removes unimportant samples and does not hurt generalization. Similar graphs for other metrics described in Table 2 can be found in Appendix B.

4.6 Characterizing Potential Failure Modes

Recent works have attempted to train classifiers on datasets that contain spurious features [23, 40] (example Waterbirds, CelebA [32] dataset). 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 the forgotten examples we can further investigate the examples that the classifier fails to generalize to.

Figure 4: By inspecting the earliest-forgotten examples, we can gain insights into potential failure modes. This model quickly forgets planes with green backgrounds and horses on blue backgrounds.

Stability of SSFT We note that SSFT is stable across multiple seeds (Pearson correlation of 0.81), and across architectures (Pearson correlation of 0.63). While the overall correlation for samples ranked by SSFT may be low across architectures, the top-ranked examples have a high correlation (0.85), suggesting the most forgotten examples are consistent across architectures. In contrast, FSLT has a Pearson correlation of 0.52 across seeds. Most interestingly, the learning time metric is brittle to the choice of hyperparameters. As shown by Jiang et al. [24], when using Adam optimizer, examples of different hardness get learned together. In our experiments, we observe the same phenomenon during learning, however, SSFT is robust to the choice of the optimizer. Detailed results in Appendix C.1.

Limitations One limitation of the proposed metric is that it is brittle to the choice of the learning rate for the second-split training. If we use a very small learning rate, then overparametrized deep models are capable of learning the new dataset without forgetting examples from the first split. Alternately, if we use a very large learning rate, the model may diverge and undergo catastrophic forgetting. However, under 'reasonable' choices of learning rate (like that for first-split training), we find SSFT is robust.

5 Theoretical Results

We consider a dataset $\mathcal{S}_A = \{\mathbf{x}_i, \mathbf{y}_i\}^n$ such that $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}$. Let $\mathbf{w} \in \mathbb{R}^d$ represent the weight vector of an overparametrized linear model. We analyze the learning and forgetting dynamics by minimizing the empirical risk: $\mathcal{L}(\mathcal{S}; \mathbf{w}) = \sum_i \ell(\mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i)$. Inspired by Soudry et al. [44], we make the following assumptions about the problem setup: (i) The dataset is linearly separable: $\exists\ \mathbf{w}_*$ such that $\mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i > 0 \ \ \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}$. (ii) $\ell(u)$ is the exponential loss. We develop theoretical arguments for the binary version of the synthetic distribution introduced in Section 4.2.

Optimization We perform gradient descent with fixed learning rate η,

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla \mathcal{L}(\mathbf{w}(t)) = \mathbf{w}(t) - \eta \sum_i \ell'(\mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i) \cdot \mathbf{y}_i \mathbf{x}_i. \tag{4}$$

Asymptotic Solution [44] For sufficiently small learning rate η, and (bounded) starting point $\mathbf{w}(0)$,

$$\mathbf{w}(t) = \hat{\mathbf{w}} \log t + \rho(t), \tag{5}$$

where $\rho(t)$ is a bounded residual term, and $\hat{\mathbf{w}}$ is the solution to the hard margin SVM:

$$\hat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}} \ \|\mathbf{w}\|^2 \ \ s.t.\ \mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i \geq 1, \tag{6}$$

Definition 1 (Rare Examples, $\mathcal{R}$ [14])**.** *Consider a dataset $\mathcal{S}$ sampled from a mixture of distributions $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ with frequency $\{\pi_1, \ldots, \pi_N\}$ respectively. Let $\mathcal{R} \subseteq \mathcal{S}$ be the set of rare examples. Then, for all $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{R}$, if $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}_j$, then there are $O(1)$ samples from $\mathcal{D}_j$ in $\mathcal{S}$.*

Definition 2 (Mislabeled Examples, $\mathcal{M}$)**.** *Consider a k class classification problem with $\mathcal{Y} = \{1, 2, \ldots, k\}$. Let $\mathcal{M} \subset \mathcal{S}$ be the set of mislabeled examples. Then for any $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, a corresponding mislabeled example is given by $(\mathbf{x}, \tilde{\mathbf{y}}) \in \mathcal{M}$ such that $\tilde{\mathbf{y}} \in \mathcal{Y} \setminus \{\mathbf{y}\}$.*[1]

Definition 3 ($(\lambda, \gamma, \sigma)$-Coupling)**.** *We parametrize any distribution $\mathcal{D}_g$ by $(\lambda_g, \gamma, \sigma)$, where for all $(\mathbf{x}_1, \mathbf{y}), (\mathbf{x}_2, \mathbf{y}) \sim \mathcal{D}_g$, we have $\mathbf{x}_1^\top \mathbf{x}_2 \sim \mathcal{N}(\lambda_g, \sigma^2)$ and for all $(\mathbf{x}_3, \mathbf{y}') \sim \mathcal{D}_{g_1}$, $(\mathbf{x}_4, \mathbf{y}'') \sim \mathcal{D}_{g_2}$, we have $\mathbf{x}_3^\top \mathbf{x}_4 \sim \mathcal{N}(\gamma, \sigma^2)$.*

Definition 4 (Complex Examples, $\mathcal{C}$)**.** *Let $\mathcal{C} \subset \mathcal{S}$ be the set of examples sampled from complex distributions. Let $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{C}$ such that $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}_g$ (complex group), then $\lambda_g = \frac{\lambda_t}{c}$, $c > 1$ for any simple distribution $\mathcal{D}_t$.*

5.1 First-split Learning

For the stage 1, we consider that we train the model for a maximum of T epochs (until we achieve 100% accuracy on the first training dataset $\mathcal{S}_A$). This means that the learned weight vectors are close to, but have not converged to the max margin solution. The solution at the end of t epochs is given by $\mathbf{w}_A(t)$. At sufficiently large T, we have:

$$\begin{aligned} \mathbf{w}_A(T) &= \hat{\mathbf{w}}_A \log T + \rho_A(T) \\ \mathbf{w}_A(T)^\top \mathbf{y}_i \mathbf{x}_i &\geq 1 \quad \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_A \end{aligned} \tag{7}$$

5.2 Second-split Forgetting

We initialize the weights for second stage of training with $\mathbf{w}_A(T)$ from first training stage, and then train on $\mathcal{S}_B$. We provide the formal theorem statement and complete proofs in Appendix A, but provide informal theorem statements and an intuitive proof sketch below:

Theorem 1 (Forgetting Dynamics (informal))**.** *For sufficiently small learning rate, given two datasets $\mathcal{S}_A, \mathcal{S}_B \sim \mathcal{D}^n$. For a model initialized with weights, $\mathbf{w}_B(0) = \mathbf{w}_A(T)$ and trained for T' = f(T) epochs, the following hold with high probability: (i) Data points that were mislabeled on the original dataset are no longer incorrectly predicted; and (ii) Rare examples are not forgotten.*

Theorem 2 (Forgetting Dynamics Asymptotic (informal))**.** *For sufficiently small learning rate, given two datasets $\mathcal{S}_A, \mathcal{S}_B \sim \mathcal{D}^n$. After training for $T' \to \infty$ epochs, the following hold with high probability: (i) Rare examples are forgotten; and (ii) Complex examples are not forgotten.*

Proof Sketch We evaluate the classification accuracy of the model for mislabeled, atypical and complex data points. More specifically, $\mathcal{S}_B$ contains similar support vectors as $\mathcal{S}_A$ for the complex examples. Hence, they stay correctly classified with high probability even when we train for long. With high probability, $\mathcal{S}_B$ does not have samples from rare groups found in $\mathcal{S}_A$. Therefore, there is no support vector in $\mathcal{S}_B$ that is able to correctly classify the rare samples of $\mathcal{S}_A$. However, the model converges slowly and is still able to classify all examples correctly with high probability until T' epochs. Finally, we analyze the case of mislabeled examples. Since, $\mathbf{w}_A$ correctly classifies them, these samples were support vectors of $\mathcal{S}_A$. However, no such support vector is present in $\mathcal{S}_B$. In our analysis, we utilize the coupling constraint (Def. 3) that implies high correlation between a mislabeled sample and support vectors from the same majority group. On the contrary, rare samples have low dot product with any other support vector, hence retaining their prediction for longer.

6 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.

[1] For binary classification, $\mathcal{Y} = \{-1, +1\}$. The labels are reversed for mislabeled examples.

References

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

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

[3] Devansh Arpit, Stanisław Jastrzębski, 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.

[4] Woonhyuk Baek. Torchskeleton. https://github.com/wbaek/torchskeleton, 2019.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

[12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[13] 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.

[14] 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.

[15] 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.

[16] Jonathan Frankle, David J Schwab, and Ari S Morcos. The early phase of neural network training. *arXiv preprint arXiv:2002.10365*, 2020.

[17] Saurabh Garg, Sivaraman Balakrishnan, Zico Kolter, and Zachary Lipton. RATT: Leveraging unlabeled data to guarantee generalization. *arXiv preprint arXiv:2105.00303*, 2021. URL https://arxiv.org/abs/2105.00303.

[18] Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's agree to agree: Neural networks share classification order on real datasets. In *International Conference on Machine Learning*, pages 3950–3960. PMLR, 2020.

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

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

[21] Jeremy Howard. Imagenette. URL https://github.com/fastai/imagenette/.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

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

[29] 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.

[30] 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.

[31] 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.

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738. IEEE Computer Society, 2015. ISBN 978-1-4673-8391-2. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2015.html#LiuLWT15.

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

[34] 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.

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

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

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

[38] 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.

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

[40] 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.

[41] 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.

[42] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[43] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

[44] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[45] 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.

[46] 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.

[47] George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*. Routledge, 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] : Our work attempts to understand the phenomenon of training and forgetting of samples in deep learning architectures. We do not see any direct concerns with ethical or social impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] : In the Appendix of the paper.
   (b) Did you include complete proofs of all theoretical results? [Yes] : In the Appendix of the paper.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] : In the Appendix
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Main paper and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] : For experiments on generalization error after example removal.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] : We use standard publicly available benchmark datasets.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code for Reproducing our experiments is provided in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We use standard publicly available benchmark datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Characterizing Datapoints via Second-Split Forgetting
Supplementary Material

A Theoretical Results

A.1 Preliminaries

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space. With $\mathcal{D}$, we define a distribution over $\mathcal{X} \times \mathcal{Y}$. In our work, we assume that $\mathcal{D}$ is a mixture of distributions $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ with $\mathcal{D} = \sum_N \pi_i \mathcal{D}_i$ where $\pi_i > 0$ for all $i \in [N]$ and $\sum_N \pi_i = 1$. Further, each $\mathcal{D}_i$ has a disjoint support with any other distribution, i.e., $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_i \implies \mathbb{P}_{\mathcal{D}_j}[(\mathbf{x}, \mathbf{y})] = 0$ for all $j \neq i$. We consider a dataset $\mathcal{S}_A = \{\mathbf{x}_i, \mathbf{y}_i\}^n$ such that $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}$. Let $\mathbf{w} \in \mathbb{R}^d$ represent the weight vector of overparametrized linear model. We analyze the learning and forgetting dynamics by minimizing the empirical risk: $\mathcal{L}(\mathcal{S}; \mathbf{w}) = \sum_i \ell(\mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i)$. Based on Soudry et al. [44], we make the following assumptions about the problem setup:

(A.1) The dataset is linearly separable: $\exists\ \mathbf{w}_*$ such that $\mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i > 0 \ \ \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}$.

(A.2) $\ell(u)$ is the exponential loss.

(A.3) The data-points sampled from the mixture of distributions $\mathcal{D}$ as define above, and $||\mathbf{x}_i|| = 1$.

For completeness, we write the definitions and preliminaries from the main paper below.

Optimization We perform gradient descent with fixed learning rate η,

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla \mathcal{L}(\mathbf{w}(t)) = \mathbf{w}(t) - \eta \sum_i \ell'(\mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i) \cdot \mathbf{y}_i \mathbf{x}_i. \tag{8}$$

Asymptotic Solution [44] For sufficiently small learning rate η, and (bounded) starting point $\mathbf{w}(0)$,

$$\mathbf{w}(t) = \hat{\mathbf{w}} \log t + \rho(t), \tag{9}$$

where $\hat{\mathbf{w}}$ is the solution to the hard margin SVM:

$$\hat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^d}{\operatorname{argmin}} \ \|\mathbf{w}\|^2 \ \ s.t.\ \mathbf{w}^\top \mathbf{y}_i \mathbf{x}_i \geq 1, \tag{10}$$

Definition 1 (Rare Examples, $\mathcal{R}$ [14])**.** *Consider a dataset $\mathcal{S}$ sampled from a mixture of distributions $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ with frequency $\{\pi_1, \ldots, \pi_N\}$ respectively. Let $\mathcal{R} \subseteq \mathcal{S}$ be the set of rare examples. Then, for all $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{R}$, if $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}_j$, then there are $O(1)$ samples from $\mathcal{D}_j$ in $\mathcal{S}$.*

Definition 2 (Mislabeled Examples, $\mathcal{M}$)**.** *Consider a k class classification problem with $\mathcal{Y} = \{1, 2, \ldots, k\}$. Let $\mathcal{M} \subset \mathcal{S}$ be the set of mislabeled examples. Then for any $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, a corresponding mislabeled example is given by $(\mathbf{x}, \tilde{\mathbf{y}}) \in \mathcal{M}$ such that $\tilde{\mathbf{y}} \in \mathcal{Y} \setminus \{\mathbf{y}\}$.*[1]

Definition 3 ($(\lambda, \gamma, \sigma)$-Coupling)**.** *We parametrize any distribution $\mathcal{D}_g$ by $(\lambda_g, \gamma, \sigma)$, where for all $(\mathbf{x}_1, \mathbf{y}), (\mathbf{x}_2, \mathbf{y}) \sim \mathcal{D}_g$, we have $\mathbf{x}_1^\top \mathbf{x}_2 \sim \mathcal{N}(\lambda_g, \sigma^2)$ and for all $(\mathbf{x}_3, \mathbf{y}') \sim \mathcal{D}_{g_1},\ (\mathbf{x}_4, \mathbf{y}'') \sim \mathcal{D}_{g_2}$, we have $\mathbf{x}_3^\top \mathbf{x}_4 \sim \mathcal{N}(\gamma, \sigma^2)$.*

In our work, we make assumptions on the coupling of datapoints within the same subgroup. This assumption is much milder than assuming the data distribution of samples as studied in prior work [31]. Further, notice that we do not assume that the datapoints from opposite classes have opposite correlation in feature space (this is often assumed in all theoretical analysis on a binary gaussian setup). Our assumption is further general since it assumes that there may exist many subgroups within the same class that may have weak correlation between them (as weak as that between two subgroups of different class). The only assumption that we make is that examples within the same subgroup have high cosine similarity, which in turn defines and is a property of their group membership.

Definition 4 (Complex Examples, $\mathcal{C}$)**.** *Let $\mathcal{C} \subset \mathcal{S}$ be the set of examples sampled from complex distributions. Let $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{C}$ such that $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}_g$ (complex group), then $\lambda_g = \frac{\lambda_t}{c}$, $c > 1$ for any simple distribution $\mathcal{D}_t$.*

[1] For binary classification, $\mathcal{Y} = \{-1, +1\}$. The labels are reversed for mislabeled examples.

The implication of the aforementioned characterization is that complex distributions have a higher signal-to-noise ratio (SNR) as compared to simple distributions. We assume the sample complexity required to estimate the distribution as a proxy for the complexity of the distribution. In this regard, having a high SNR increases the complexity.

Based on the above definitions, we assign the same intra-group coupling mean for all simple sub-distributions (λ) in $\mathcal{D}$, and a smaller (but same) intra-group coupling mean for all complex distributions (λ/c). With this, we parametrize our mixture of distributions $\mathcal{D}$ by $(\lambda, \gamma, \sigma, c)$ where λ is the coupling mean for any simple sub-distribution, and λ/c is the coupling mean for complex distributions. Coupling between any two sub-distributions is parametrized by the same mean γ.

A.2 Learning Stage

For the stage 1, we consider that we train the model for a maximum of T epochs (until we achieve 100% accuracy on the first training dataset $\mathcal{S}_A$ with margin greater than 1). This means that the learned weight vectors are close to, but have not converged to the max margin solution. The solution at the end of T epochs is given by $\mathbf{w}_A(t)$. At sufficiently large T, we have:

$$\begin{aligned} \mathbf{w}_A(T) &= \hat{\mathbf{w}}_A \log T + \rho_A(T) \\ \mathbf{w}_A(T)^\top \mathbf{y}_i \mathbf{x}_i &\geq 1 \quad \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_A \end{aligned} \tag{11}$$

Proposition 1 (Bounded Weights)**.** *There exists a bounded time t beyond which the model classifies all training points correctly.*

Proof. From the assumptions made, we know that the dataset is separable. This means that the max-margin solution (or the SVM solution) for the dataset classifies all training points correctly. From the analysis in Soudry et al. [44], we know that:

$$\begin{aligned} \lim_{t\to\infty} \frac{\mathbf{w}_A(t)}{||\mathbf{w}_A(t)||} &= \frac{\hat{\mathbf{w}}_A(t)}{||\hat{\mathbf{w}}_A(t)||}, \\ \lim_{t\to\infty} \mathbf{y}_i \mathbf{x}_i^\top \frac{\mathbf{w}_A(t)}{||\mathbf{w}_A(t)||} &= \mathbf{y}_i \mathbf{x}_i^\top \frac{\hat{\mathbf{w}}_A(t)}{||\hat{\mathbf{w}}_A(t)||} \end{aligned} \tag{12}$$

Then it directly follows that, for $\epsilon > 0$, there exists bounded time $T > 0$ such that $\forall t > T$,

$$\left| \mathbf{y}_i \mathbf{x}_i^\top \frac{\mathbf{w}_A(t)}{||\mathbf{w}_A(t)||} - \mathbf{y}_i \mathbf{x}_i^\top \frac{\hat{\mathbf{w}}_A(t)}{||\hat{\mathbf{w}}_A(t)||} \right| < \epsilon$$

This concludes that there exists a time T for which the sign of the prediction of both the max-margin solution and the learnt solution will be the same, implying correct prediction for every example in the training set for a bounded time solution.

□

A.3 Forgetting Stage

We initialize the weights for second stage of training with $\mathbf{w}_A(T)$ from first training stage, and then train on $\mathcal{S}_B$ to minimize the empirical loss using gradient descent (Equation 4). Assume that the dataset is balanced in the class labels, $|\mathcal{S}_B| = 2n$. Also, for simplicity assume that $\mathcal{S}_B$ does not contain mislabeled or rare examples from the same sub-group as in $\mathcal{S}_A$.

A.4 Asymptotic Analysis

We will first provide a formal version of Theorem 2 which was informally stated in the main paper.

Theorem 3 (Asymptotic Forgetting)**.** *For a mixture of distributions partametrized by $(\lambda, \gamma, \sigma, c)$, under assumptions A.1, A.2, A.3, as we fine-tune for $t \to \infty$ iterations on the second dataset $\mathcal{S}_B$, $(\mathbf{x}_m, \mathbf{y}_m) \in \mathcal{M} \subset \mathcal{S}_A$ is correctly classified with probability $\Phi\left(\frac{1}{\sqrt{2}} - \lambda/\sigma\right)^{O(n)}$, $(\mathbf{x}_r, \mathbf{y}_r) \in \mathcal{R} \subset \mathcal{S}_A$ is misclassified with probability 0.5 and $(\mathbf{x}_c, \mathbf{y}_c) \in \mathcal{C} \subset \mathcal{S}_A$ is misclassified with probability $\Phi\left(\left(\frac{1}{\sqrt{2}} - \frac{\lambda}{c}\right)/\sigma\right)^{O(n)}$.*

First, we analyze infinite-time training case. We will extend the result from Soudry et al. [44] to show that (i) mislabeled and rare examples are forgotten when the model is trained for long; and (ii) complex examples are not forgotten. First, recall the result used in Subsection A.2 for any bounded initialization for weights $\mathbf{w}_B(0)$,

$$\mathbf{w}_B(t) = \hat{\mathbf{w}}_B \log t + \rho_B(t). \tag{13}$$

Recall the characterization of samples based on the subgroups that they belong to. As in definition 3, we parametrize any distribution $\mathcal{D}_g$ by $(\lambda_g, \gamma, \sigma)$, where $\forall (\mathbf{x}_1, \mathbf{y}_g), (\mathbf{x}_2, \mathbf{y}_g) \sim \mathcal{D}_g$, we have $\mathbf{x}_1^\top \mathbf{x}_2 \sim \mathcal{N}(\lambda_g, \sigma^2)$ and $\forall (\mathbf{x}_3, \mathbf{y}_{g_1}) \sim \mathcal{D}_{g_1},\ (\mathbf{x}_4, \mathbf{y}_{g_2}) \sim \mathcal{D}_{g_2}$, we have $\mathbf{x}_3^\top \mathbf{x}_4 \sim \mathcal{N}(\gamma, \sigma^2)$. As stated in the problem setup, we assume that the mislabeled sample belongs to a majority class. Without loss of generality, let the distribution be $\mathcal{D}_1$. Therefore, both $\mathcal{S}_A$ and $\mathcal{S}_B$ contain $O(n)$ samples from $\mathcal{D}_1$. On the contrary, we assume that $\mathcal{S}_A$ contains only one sample from $\mathcal{D}_r$, and no such sample in $\mathcal{S}_B$.

Mislabeled Examples Since $\mathcal{S}_B$ is separable, in the limit of infinite-training time, the classifier correctly predicts all the examples in the dataset. We will use this fact to show that it assigns the opposite label to the mislabeled sample in set $\mathcal{S}_A$ with high probability. Let there be $n_1 = O(n)$ samples from majority group $\mathcal{D}_1$ to which our mislabeled sample $(\mathbf{x}_m, \mathbf{y}_m)$ belongs to. Let $\mathcal{S}_{B,1}$ represent the set of examples in the second-split which are sampled from the distribution $\mathcal{D}_1$. Then, we denote 'failure' as the event that the mislabeled sample is still predicted with its assigned label at infinite-time training. We will show that this probability tends to 0. Without loss of generality, we assume that the true label of all samples from $\mathcal{D}_1$ is $+1$, and hence $\mathbf{y}_m = -1$ (since it was mislabeled). Denote 'failure' as the event that the mislabeled samples is still predicted with label $\mathbf{y}_m$.

$$\begin{aligned}
\mathbb{P}\left[\text{failure}\right] &= \mathbb{P}\left[\mathbf{x}_m^\top \mathbf{w} < 0 \mid \mathbf{x}_i^\top \mathbf{w} \geq 1;\ \mathbf{x}_i^\top \mathbf{x}_m \sim \mathcal{N}(\lambda_1, \sigma^2)\ \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_{B,1}\right], \\
\mathbf{x}_m &= \mathbf{x}_i \pm \mathbf{x}_i \frac{\sqrt{1 - z_i^2}}{z_i}\ s.t\ z_i = \frac{v_i}{||\mathbf{x}_m|| \cdot ||\mathbf{x}_i||}\ ;\ v_i\ \sim \mathcal{N}(\lambda_1, \sigma^2) \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_{B,1}, \\
\mathbf{x}_m^\top \mathbf{w} &= \mathbf{x}_i^\top \mathbf{w} \pm \mathbf{x}_i^\top \mathbf{w} \frac{\sqrt{1 - z_i^2}}{z_i}, \\
\mathbf{x}_m^\top \mathbf{w} &= \mathbf{x}_i^\top \mathbf{w} \left(1 \pm \frac{\sqrt{1 - z_i^2}}{z_i}\right), \\
\text{sign}(\mathbf{x}_m^\top \mathbf{w}) &= 1 \pm \frac{\sqrt{1 - z_i^2}}{z_i}\ \ \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_{B,1}.
\end{aligned} \tag{14}$$

Hence,

$$\begin{aligned}
\mathbb{P}\left[\mathbf{x}_m^\top \mathbf{w} < 0 \mid \text{conditions}\right] &\leq \mathbb{P}\left[-\frac{1}{\sqrt{2}} < z_i < \frac{1}{\sqrt{2}}\ \forall z_i\right] \\
&\leq \mathbb{P}\left[z_i < \frac{1}{\sqrt{2}}\right]^{n_1} \\
&\leq \mathbb{P}\left[v_i < \frac{1}{\sqrt{2}}\ ;\ v_i\ \sim \mathcal{N}(\lambda_1, \sigma^2)\right]^{n_1}, \\
&\leq \Phi\left(\frac{\frac{1}{\sqrt{2}} - \lambda_1}{\sigma}\right)^{n_1}
\end{aligned} \tag{15}$$

where n_1 is $O(n)$ since the mislabeled sample belongs to a minority group. The probability of which tends to 0 whenever the coupling effect is strong ($\lambda_1 > \frac{1}{\sqrt{2}}$, or the angle between the vectors is less than $45°$). Now substitute $\lambda_1 = \lambda$ to get the desired result.

Rare Examples We will utilize the fact that rare examples have negligible positive coupling with any other example in the second-split dataset $\mathcal{S}_B$. Without loss of generality, let us assume that the rare samples has label $+1$. Using the assumption of the dataset being balanced, there are n samples that

have label $+1$ and have γ-coupling with $\mathbf{x}_r$, and n samples that have label -1, and have γ-coupling with $\mathbf{x}_r$. Let us specify the above as 'conditions' = $\{\mathbf{x}_i^\top \mathbf{w} \geq 1;\ \mathbf{x}_i^\top \mathbf{x}_m \sim \mathcal{N}(\gamma, \sigma^2)\ \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_B^+\} \bigcup \{\mathbf{x}_i^\top \mathbf{w} \leq -1;\ \mathbf{x}_i^\top \mathbf{x}_m \sim \mathcal{N}(\gamma, \sigma^2)\ \forall (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_B^-\}$.

Then, from symmetry argument, we have that:

$$\mathbb{P}\left[\mathbf{y}_r \mathbf{x}_r^\top \mathbf{w} > 0 \mid \text{conditions}\right] = \mathbb{P}\left[\mathbf{y}_r \mathbf{x}_r^\top \mathbf{w} < 0 \mid \text{conditions}\right] = 0.5 \tag{16}$$

Note that this analysis also highlights that the meaning of being 'forgotten' for rare examples is to predict randomly. However, in case of mislabeled examples, the predicted label of the example approaches its 'true' label, irrespective of its training label.

Complex Examples The analysis for complex examples follows directly from the analysis in the case of mislabeled examples. The only difference is the strength of the coupling within the group. Recall that complex examples are also sampled from majority groups. Without loss of generality we can refer the corresponding distribution as $\mathcal{D}_2$, with coupling constant λ_2, and the subset of examples in the second-split as $\mathcal{S}_{B,2}$, with true label $+1$. Therefore, the probability that the complex example is mis-classified at the end of training for infinite time is given by:

$$\mathbb{P}\left[\mathbf{x}_c^\top \mathbf{w} < 0 \mid \text{conditions}\right] \leq \Phi\left(\frac{\frac{1}{\sqrt{2}} - \lambda_2}{\sigma}\right)^{n_2} \tag{17}$$

This once again approaches 0, indicating perfect classification of complex examples from $\mathcal{S}_A$. Now substitute $\lambda_2 = \frac{\lambda}{c}$ to get the desired theorem result.

A.5 Intermediate Time Analysis

From the analysis in Section A.4, we find that all the mislabeled and rare examples are forgotten by the time we train for $t \to \infty$ iterations. Since complex examples are not forgotten even at infinite time training, we skip analysis for those examples in this subsection. Our goal is to show that there exists a bounded time T' when the model forgets all the mislabeled examples with high probability, but still correctly classifies all the rare examples with high probability. To show this, we will track the accuracy for inputs in the first data split, as we train on the examples in the second split.

The model output for any sample $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}_A$ is given by $\mathbf{w}(t)^\top \mathbf{x}_i$, and the prediction is considered to be correct of $\text{sign}(\mathbf{w}(t)^\top \mathbf{x}_i) = \mathbf{y}_i$, or if $\mathbf{a}_i^t = \mathbf{w}(t)^\top \mathbf{x}_i \mathbf{y}_i > 0$. From hereon, we will use the notation $\mathbf{a}_i^t = \mathbf{w}(t)^\top \mathbf{x}_i \mathbf{y}_i$.

To simplify our analysis, we consider that there is one example from both the mislabeled and rare example category in $\mathcal{S}_A$. We will denote these data points by $(\mathbf{x}_m, \mathbf{y}_m) \in \mathcal{M}$, and $(\mathbf{x}_r, \mathbf{y}_r) \in \mathcal{R}$ respectively. Without loss of generality, we will assume that $(\mathbf{x}_m, \mathbf{y}_m)$ was sampled from $\mathcal{D}_1$ in the mixture of distributions $\mathcal{D}$. All the examples in the second split $\mathcal{S}_B$ sampled from the same distribution are given by $\mathcal{S}_{B,1} \subset \mathcal{S}_B$. The remaining examples are in the set denoted by $\mathcal{S}_{B,\neg 1} \subset \mathcal{S}_B$.

From the learning time training dynamics, we know that the initialization of the weights for the second round of training is given by:

$$\mathbf{w}_B(0) = \hat{\mathbf{w}}_A \log T + \rho_A(T). \tag{18}$$

Now, from the representer theorem, we can decompose the model weights at any iteration of second-split training as follows,

$$\mathbf{w}_B(t) = \hat{\mathbf{w}}_A \log T + \rho_A(T) + \sum_{j \in \mathcal{S}_B} \beta_j \mathbf{y}_j \mathbf{x}_j. \tag{19}$$

Note that we introduce an additional term $\mathbf{y}_j$ in the decomposition using representer theorem, but this can be done without loss of generality since $\mathbf{y}_j \in \{-1, +1\}$. This helps us in ensuring that each β_j is non-negative as shown in Lemma 1.

Lemma 1 (Sign of β)**.** *$\beta_j \geq 0$ for all j in Equation 19.*

Proof. Analyzing the steps of gradient descent, we have:

$$\begin{aligned}\dot{\mathbf{w}}(t) &= -\nabla \mathcal{L}(\mathbf{w}(t)) \\ &= \sum_{j \in \mathcal{S}_B} \exp\left(-\mathbf{y}_j \mathbf{x}_j^\top \mathbf{w}(t)\right)(\mathbf{y}_j \mathbf{x}_j^\top) \\ \mathbf{w}(t) - \mathbf{w}(0) &= \sum_{j \in \mathcal{S}_B} \left((\mathbf{y}_j \mathbf{x}_j^\top) \underbrace{\int_0^t \exp\left(-\mathbf{y}_j \mathbf{x}_j^\top \mathbf{w}(t)\right) dt}_{\beta_j(t)} \right),\end{aligned} \tag{20}$$

Hence, $\beta_j \geq 0 \; \forall \; j$. □

Let us define $\delta_t = \frac{\sum_{j \in \mathcal{S}_{B,1}} \beta_j(t)}{\sum_{j \in \mathcal{S}_B} \beta_j(t)}$, and $\delta = \max_t \delta_t$. For a symmetric distribution with two majority subgroups with the opposite label, we would expect this value to be close to 0.5. Now, we present our formal theorem statement. We will now provide a formal version of Theorem 1 which was informally stated in the main paper.

Theorem 4 (Intermediate-Time Forgetting)**.** *For a mixture of distributions parametrized by $(\lambda, 0, \sigma, c)$, under assumptions A.1, A.2, A.3, there exists some bounded time T when $(\mathbf{x}_m, \mathbf{y}_m) \in \mathcal{M} \subset \mathcal{S}_A$ is misclassified with probability $\Phi(\lambda\delta/2\sigma)$ and $(\mathbf{x}_r, \mathbf{y}_r) \in \mathcal{R} \subset \mathcal{S}_A$ is correctly classified with the same probability when fine-tuning on the second dataset split $\mathcal{S}_B$ with sufficiently small learning rate.*

Lemma 2 (Support Vectors)**.** *$(\mathbf{x}_m, \mathbf{y}_m)$ and $(\mathbf{x}_r, \mathbf{y}_r)$ are support vectors for the dataset $\mathcal{S}_A$.*

Proof. We will prove by contradiction. From our assumption, $(\mathbf{x}_m, \mathbf{y}_m)$ and $(\mathbf{x}_r, \mathbf{y}_r)$ are the only mislabeled and rare examples in the first-split $\mathcal{S}_A$ from their respective sub-groups. Let us assume that they are not support vectors. Then, we can directly follow from the Asymptotic Analysis in Subsection A.4 that the probability of correct classification of the rare example is 0.5 and for the mislabeled example approaches 0 as the model is trained for infinite time. But we know that the model achieves 100% accuracy on the training set $\mathcal{S}_A$ with weights $\mathbf{w}_A(T)$. Hence, this is a contradiction, and $(\mathbf{x}_m, \mathbf{y}_m)$, $(\mathbf{x}_r, \mathbf{y}_r)$ must be support vectors for the original classification problem. □

Lemma 3 (Bounded Time Prediction)**.** *There exists a bounded time T' = O(T) when the mislabeled examples are incorrectly classified with high probability.*

Proof. In Subsection A.4 we have shown that at infinite time training, the model misclassifies mislabeled examples with high probability. Then, the existence of bounded time weights for which the prediction of mislabeled examples is flipped directly follows from the proof of Proposition 1. □

In what follows, the key idea is to show that at a time when the prediction of the mislabeled example is incorrect with high probability, the predictions of the rare examples is correct with high probability.

From Lemma 2, we know that $(\mathbf{x}_m, \mathbf{y}_m)$ and $(\mathbf{x}_r, \mathbf{y}_r)$ are support vectors for $\mathcal{S}_A$. Hence, $\mathbf{y}_m \mathbf{x}_m^\top \hat{\mathbf{w}}_A = \mathbf{y}_r \mathbf{x}_r^\top \hat{\mathbf{w}}_A = 1$. We will use this fact, together with the coupling relation introduced in the definitions to find the distribution for $\mathbf{a}_m^t, \; \mathbf{a}_r^t$.

Mislabeled Example The accuracy on the mislabeled point can be simplified as follows:

$$\begin{aligned}\mathbf{a}_m^t = \mathbf{y}_m \mathbf{x}_m^\top \mathbf{w}_B(t) &= \mathbf{y}_m \mathbf{x}_m^\top (\hat{\mathbf{w}}_A \log T + \rho_A(T) + \sum_{j \in \mathcal{S}_B} \beta_j \mathbf{y}_j \mathbf{x}_j), \\ &= \log T + c_m + \sum_{j \in \mathcal{S}_B} \beta_j (\mathbf{y}_m \mathbf{x}_m^\top \mathbf{y}_j \mathbf{x}_j).\end{aligned} \tag{21}$$

Let $v_j \sim \mathcal{N}(\lambda, \sigma^2)$ and $u_j \sim \mathcal{N}(\gamma, \sigma^2)$

$$\begin{aligned}\sum_{j\in\mathcal{S}_B} \beta_j(\mathbf{y}_m\mathbf{x}_m^\top\mathbf{y}_j\mathbf{x}_j) &= \sum_{j\in\mathcal{S}_{B,1}} \beta_j(\mathbf{y}_m\mathbf{x}_m^\top\mathbf{y}_j\mathbf{x}_j) + \sum_{j\in\mathcal{S}_{B,\neg 1}} \beta_j(\mathbf{y}_m\mathbf{x}_m^\top\mathbf{y}_j\mathbf{x}_j) \\ &= -\sum_{j\in\mathcal{S}_{B,1}} \beta_j(\mathbf{x}_m^\top\mathbf{x}_j) + \sum_{j\in\mathcal{S}_{B,\neg 1}} \beta_j(\mathbf{y}_m\mathbf{x}_m^\top\mathbf{y}_j\mathbf{x}_j), \\ &= -\sum_{j\in\mathcal{S}_{B,1}} \beta_j(v_j) + \sum_{j\in\mathcal{S}_B^-} \beta_j(u_j) - \sum_{j\in\mathcal{S}_B^+\setminus\mathcal{S}_{B,1}} \beta_j(u_j),\end{aligned} \tag{22}$$

In the last step we split the dataset into positive and negative samples, assuming a balanced dataset with n samples with label $+1$ and -1 each. Since $\gamma = 0$,[2]

$$\mathbf{a}_m^t = \log T + c_m + z_m;\ \text{s.t. } z_m \sim \mathcal{N}(-\sum_{j\in\mathcal{S}_{B,1}} \beta_j\lambda, \sum_{j\in\mathcal{S}_B} \beta_j^2\sigma^2). \tag{23}$$

Rare Example Following the analysis of the mislabeled example, we can similarly find the accuracy on the rare example as follows:

$$\begin{aligned}\mathbf{a}_r^t = \mathbf{y}_r\mathbf{x}_r^\top\mathbf{w}_B(t) &= \mathbf{y}_r\mathbf{x}_r^\top(\hat{\mathbf{w}}_A\log T + \rho_A(T) + \sum_{j\in\mathcal{S}_B} \beta_j\mathbf{y}_j\mathbf{x}_j), \\ &= \log T + c_r + \sum_{j\in\mathcal{S}_B} \beta_j(\mathbf{y}_r\mathbf{x}_r^\top\mathbf{y}_j\mathbf{x}_j),\end{aligned} \tag{24}$$

The rare example has a low coupling (γ) with all the examples in $\mathcal{S}_B$. Once again decomposing the last term,

$$\begin{aligned}\sum_{j\in\mathcal{S}_B} \beta_j(\mathbf{y}_r\mathbf{x}_r^\top\mathbf{y}_j\mathbf{x}_j) &= \sum_{j\in\mathcal{S}_{B,1}} \beta_j(\mathbf{y}_m\mathbf{x}_m^\top\mathbf{y}_j\mathbf{x}_j) + \sum_{j\in\mathcal{S}_{B,\neg 1}} \beta_j(\mathbf{y}_m\mathbf{x}_m^\top\mathbf{y}_j\mathbf{x}_j) \\ &= -\sum_{j\in\mathcal{S}_B^-} \beta_j(u_j) + \sum_{j\in\mathcal{S}_B^+} \beta_j(u_j)\end{aligned} \tag{25}$$

Since $\gamma = 0$,

$$\mathbf{a}_r^t = \log T + c_r + z_r,\ \text{s.t. } z_r \sim \mathcal{N}(0, \sum_{j\in\mathcal{S}_B} \beta_j^2\sigma^2). \tag{26}$$

Combining both cases Let $\alpha = \sum_{j\in\mathcal{S}_B} \beta_j$ and $\alpha_1 = \sum_{j\in\mathcal{S}_{B,1}} \beta_j$. Based on the analysis of Soudry et al. [44], we know that $\mathbf{w}_B(t)$ grows as fast as $O(\log t)$. Therefore, both $\alpha_1 = O(\log t)$, and $\alpha = O(\log t)$ (must grow at most as fast as that). From the problem definition, $\alpha_1 = \delta\alpha$, where $\delta \in [0, 1]$. Further, use the fact that $\alpha^2 = (\sum_{j\in\mathcal{S}_B} \beta_j)^2 > \sum_{j\in\mathcal{S}_B} \beta_j^2$ since every $\beta_j > 0$.

We know that $\mathbf{a}_m^t < 0$ with high probability. Consider the 3-σ confidence interval for $\mathbf{a}_m^t$ to be below 0. Then, $\mathbf{a}_m^t = \log T + c_m - \delta\alpha\lambda \pm 3\alpha\sigma < 0$. Note that the theorem assumes a sufficiently small learning rate, which ensure that there exists an iteration when $\mathbf{a}_m^t$ belongs to the range described above.

Now, we will construct the 3σ confidence interval for $\mathbf{a}_r^t$ and find the probability of it lying above 0. $\mathbf{a}_r^t = \log T + c_r \pm 3\alpha\sigma$. Therefore, as long as $\lambda > \frac{6\sigma}{\delta}$, we have that $\mathbf{a}_r^t > 0$ with probability $\Phi(3) \sim 0.9986$ (one sided value for 3σ confidence). Similarly, we can substitute 3 with $k/2$ to find the probability of lying in any $\frac{k}{2}\sigma$ confidence interval. This concludes the proof of Theorem 4.

B Experimental Results

B.1 Experimental Setup

Architectures We perform experiments using four different model architectures—LeNet, ResNet-9 [4], ResNet-50, and Bert-base-cased [13]. 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.

[2] The analysis can be directly extended for non-zero values of γ, but requires more book-keeping

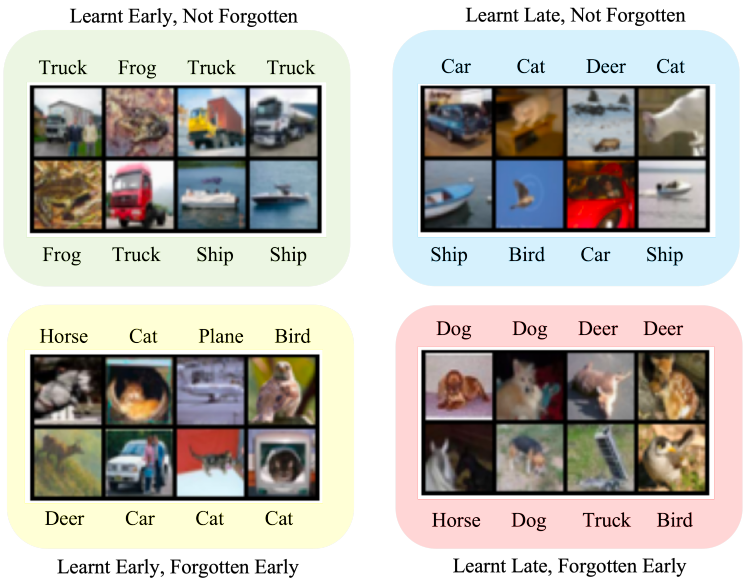


Figure 5: Examples from the CIFAR-10 dataset grouped based on their learning and forgetting time.

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.

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 [14]. 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 based on singleton correlation alone.

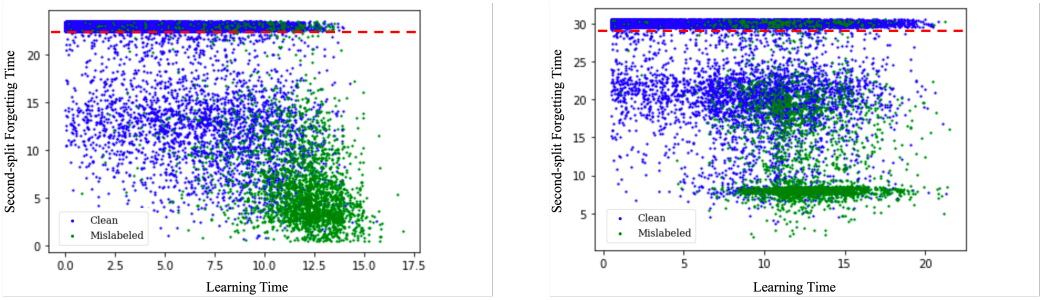
Hence, in order to distill the frequency of occurrence of an example with other confounders that may influence its training-time, we created a long-tailed dataset from the CIFAR-100 dataset. CIFAR-100 is a dataset of 100 object classes, which can be further grouped into 20 super-classes. For instance, 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 [41] 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. [41] 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.



(a) (b)

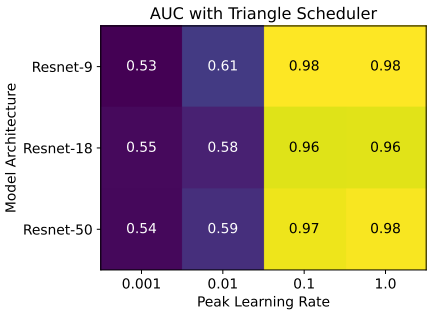
Figure 6: FSLT (First-split learning time) is able to provide some degree of separation between mislabeled and clean samples when trained with the SGD optimizer (left), but fails when the model is trained using Adam (right) on the CIFAR-10 dataset.

Stability across optimizers Jiang et al. [24] 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.

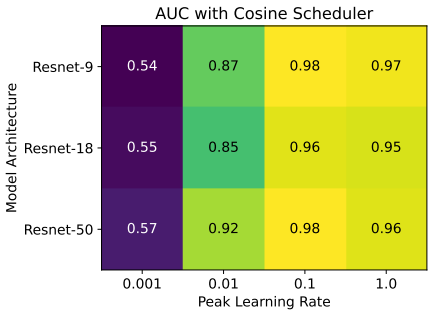
Stability across seeds and learning rates The pearson correlation for stability across seeds for the forgetting time metric is 0.83. This is higher than the corresponding learning time based metric (correlation 0.56). However, one of the drawbacks of our proposed metric is that the SSFT requires the 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.

Stability Across Learning Rate Schedules We experiment with three different learning rate schedules for the second-split training—triangular, cosine and linear. In triangular learning rate, we increase the learning rate from 0 to the maximum set value linearly over the first 10 epochs, and then decay it back to 0 until we reach the last epoch (maximum of 100 epochs). In case of the linear schedule, we increase the learning rate from 0 to the maximum set value linearly over the course of 100 epochs. The intuition behind using a linear learning rate schedule was to be able to set higher peak learning rates so that the model eventually forget all the examples in the first split and we can create a better ordering between samples based on forgetting time (as opposed to the setting where only a small fraction of examples are ever forgotten).

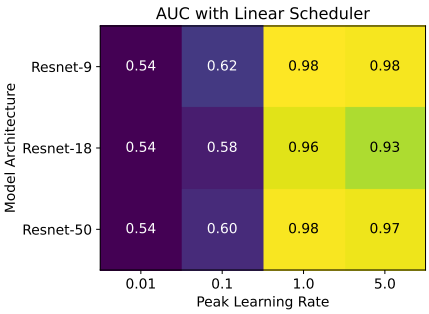
The results of the combined analysis across hyperparameters such as architecture, learning rate and learning rate schedule are presented in the heatmap of AUC of mislabeled example detection in Figure 7. The experiment was performed on the CIFAR-10 dataset with 10% label noise, and the forgetting times were averaged over 5 seeds before using them for AUC calculation. We can see that uniformly across architectures and learning rate schedules having a very low learning rate makes nearly all examples indistinguishable based on forgetting time. This is because all the models are sufficiently overparametrized to memorize all the examples in the dataset. Hence, when using a very small learning rate the optimization step moves the model weights insignificantly and we do not forget many mislabeled samples from the first split. On the other side, having a large learning rate helps achieve a strong separation between mislabeled and clean examples which also shows up in the form of high AUC values in the figure.



(a) Triangular Learning Schedule



(b) Cosine Learning Schedule



(c) Linear Learning Schedule

Figure 7: We present a heatmap for the AUC of detection of mislabeled examples using the SSFT metric under various learning rates, architecture sizes and learning rate schedules for the SGD optimizer. The experiment was performed on the CIFAR-10 dataset with 10% label noise, and the forgetting times were averaged over 5 seeds before using them for AUC calculation.

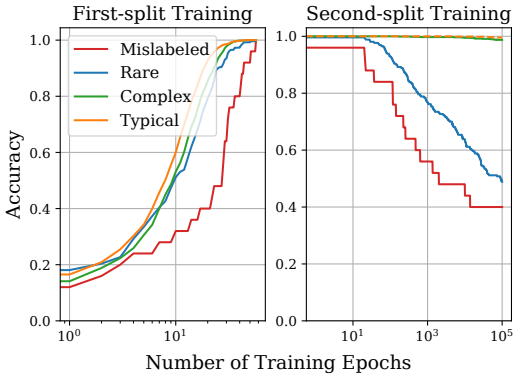


Figure 8: Learning and forgetting curves for mislabeled and rare examples when the mislabeled examples are drawn from rare subgroups in the synthetic setup described in Section 4.2.

C.2 Impact of Sampling Frequency of Mislabeled Examples

In the synthetic experiment performed in Section 4.2 we assumed that mislabeled examples occur from the majority subgroups. As a result, we observed that they get forgotten quickly during second-split training. However, in this section we aim to understand the impact of sampling frequency on the forgetting time of mislabeled examples. More specifically, we now assume that mislabeled samples occur in rare subgroups in the synthetic setup. We find that the learning curves of the mislabeled example stays the same as before, but the forgetting time for mislabeled examples closely approaches that of rare examples. This is because there is very little signal for the model to learn the opposite class during second split training since the example occurs only O(1) times. The learning and forgetting

curves pertaining to the same experiment are presented in Figure 8. In contrast with the forgetting curve in Section 4.2 where the mislabeled examples are quickly forgotten and their prediction is flipped, we find that when the subgroup corresponding to the mislabeled examples is infrequent, their forgetting time closely corresponds to that of rare examples; and on aggregate their predictions do not get flipped in the epochs that the model was trained for.