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 examples 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 also can help to 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 3.1 (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 3.2 (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].

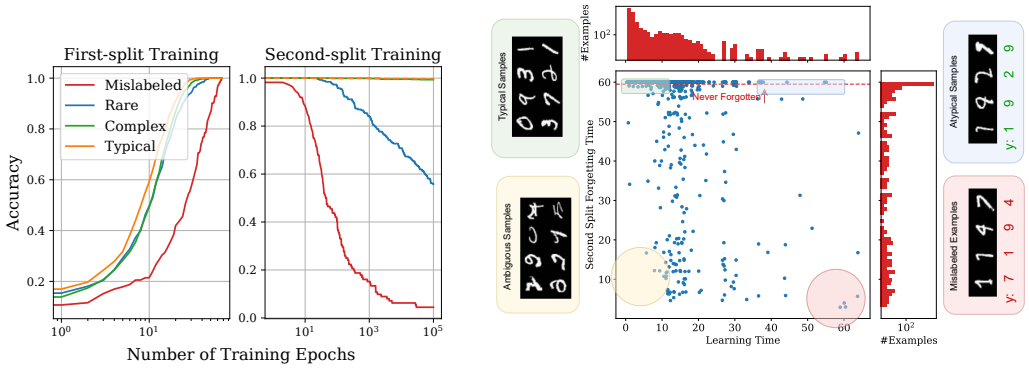
Number of Forgetting Events: (n_f). An example $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}$ undergoes a forgetting event when the accuracy on the example decreases between two consecutive updates. Toneva et al. [45] analyzed the total number of such events n_f during the training of a neural network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

Cumulative Learning Accuracy: (acc_l). Jiang et al. [24] suggest that rather than using the learning time (Definition 3.1), 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.

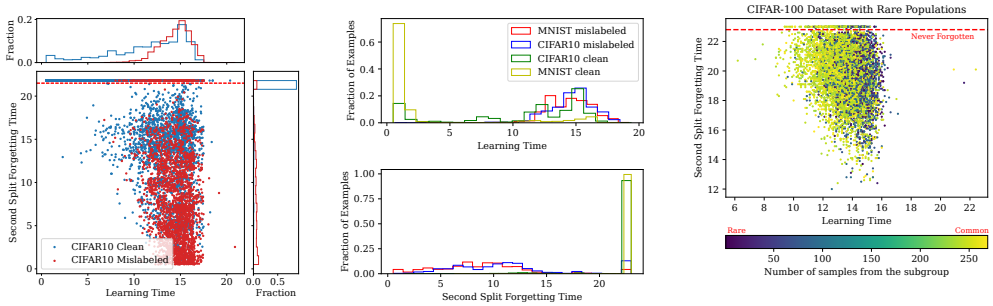
Training Details Unless otherwise specified, we train a ResNet-9 model [4] using SGD optimizer with weight decay 5e-4 and momentum 0.9. We use the cyclic learning rate schedule [42] with a peak learning rate of 0.1 at the 10th epoch. We train for a maximum of 100 epochs or until we have 5 epochs of 100% 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 $\mathcal{I}_i \cap \mathcal{I}_j = \phi$. The discriminative characteristic of each group is the vector $\mathbf{u}_g$, such that,

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.

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

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

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.

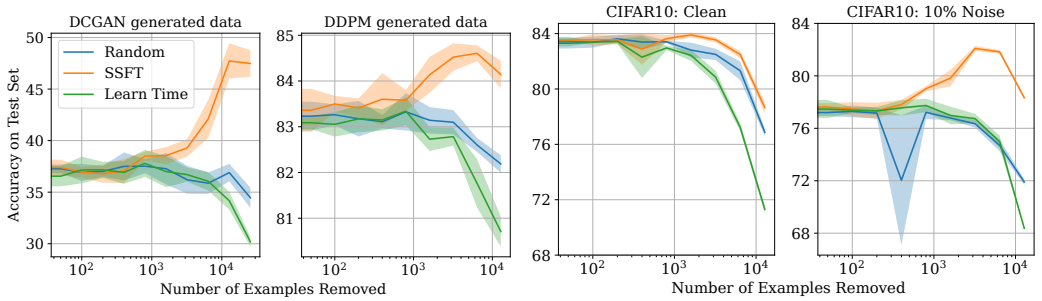
Complex Examples We generate a joint dataset that contains the union of both MNIST and CIFAR-10 examples. This is motivated by work in simplicity bias [41] that argue that neural networks learn simpler features first. We also add 10% labeled noise to each of the datasets in the union to understand the 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 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: (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. Then, we develop theoretical arguments for the binary version of the synthetic distribution $\mathcal{D} \equiv \{\mathcal{D}_i, \pi_i\}_1^N$ 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 $\hat{\mathbf{w}}$ 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 5.1 (Rare Examples, $\mathcal{R}$ [14])**.** *Let $\mathcal{S}$ be sampled from a mixture of distributions $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ with frequency $\{\pi_1, \ldots, \pi_N\}$ respectively. Then, $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{R} \subset \mathcal{S}$, if $(\mathbf{x}_i, \mathbf{y}_i) \sim \mathcal{D}_j$ and there are $O(1)$ samples from $\mathcal{D}_j$ in $\mathcal{S}$.*

Definition 5.2 (Mislabeled Examples, $\mathcal{M}$)**.** *Let $\mathcal{D} \equiv (\mathcal{X}, \mathcal{Y})$. $\mathcal{Y} \equiv \{1, 2, \ldots, k\}$ for any k class classification problem. Then for any $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, a corresponding noisy example is given by $(\mathbf{x}, \tilde{\mathbf{y}}) \in \mathcal{M} \subset \mathcal{S}$ such that $\tilde{\mathbf{y}} \sim \mathcal{Y}, \tilde{\mathbf{y}} \neq \mathbf{y}$. When $\mathcal{Y} \equiv \{-1, +1\}$ labels are reversed.*

Definition 5.3 ($(\lambda, \gamma, \sigma)$-Coupling)**.** *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 $\mathbf{x}_1 \cdot \mathbf{x}_2 \sim \mathcal{N}(\lambda_g, \sigma^2)$ and $\forall (\mathbf{x}_1, \mathbf{y}_{g_1}) \sim \mathcal{D}_{g_1},\ (\mathbf{x}_2, \mathbf{y}_{g_2}) \sim \mathcal{D}_{g_2}$, we have $\mathbf{x}_1 \cdot \mathbf{x}_2 \sim \mathcal{N}(\gamma, \sigma^2)$*

Definition 5.4 (Complex Examples, $\mathcal{C}$)**.** *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 $\mathcal{D}_t$ ($\mathcal{D}_g$ has high signal to noise ratio).*

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 5.5 (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 5.6 (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. 5.3) 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.

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]