Bandit-Driven Batch Selection for Robust Learning under Label Noise

Michal Lisicki MLISICKI@UOGUELPH.CA
University of Guelph, Vector Institute

Mihai Nica NICAM@UOGUELPH.CA
University of Guelph, Vector Institute

Graham W. Taylor GWTAYLOR@UOGUELPH.CA
University of Guelph, Vector Institute

Abstract

We introduce a novel approach for batch selection in Stochastic Gradient Descent (SGD) training, leveraging combinatorial bandit algorithms. Our methodology focuses on optimizing the learning process in the presence of label noise, a prevalent issue in real-world datasets. Experimental evaluations on the CIFAR-10 dataset reveal that our approach consistently outperforms existing methods across various levels of label corruption. Importantly, we achieve this superior performance without incurring the computational overhead commonly associated with auxiliary neural network models. This work presents a balanced trade-off between computational efficiency and model efficacy, offering a scalable solution for complex machine learning applications.

1. Introduction

As applications increasingly demand larger and more complex deep learning models, the need for efficient training strategies has become paramount. One way to accelerate training and potentially improve model performance is through the use of Curriculum Learning (CL) and adaptive batch selection. These techniques optimize learning by selectively focusing on data samples that are intrinsically rich and informative at the most appropriate stages of the learning process. Such strategies not only accelerate convergence but also enhance the model's ability to generalize [15, 16, 21].

While many methods use difficulty metrics to select easy, hard, or uncertain instances for training [23], a key area lies in handling noisy or mislabeled datasets [22]. This domain is particularly important for two reasons: a) the impact of batch selection strategies is easily measured, leading to more insightful conclusions; and b) it addresses the prevalent real-world scenarios where data is often sourced from the web [14] or crowdsourced [5], and a large portion is considered 'unclean'.

Sample selection strategies using auxiliary Deep Neural Networks (DNN) effectively mitigate the impact of noisy or mislabeled data. However, these approaches incur substantial computational overhead, limiting their scalability [7, 11, 13, 24]. While alternative methods like SELFIE [20] offer computational efficiency, they are under-explored and rely on steps like relabeling for optimal performance. Meanwhile, the literature on CL and batch selection offers numerous methods for efficient sample selection across diverse domains [6, 15].

This paper introduces a novel approach that synergizes insights from the CL and batch selection literature to enhance efficient sampling schemes, specifically targeting scenarios with prevalent label noise. Our methodology aims to achieve superior performance without the computational burden associated with deploying additional DNNs, thereby balancing efficacy and computational efficiency. Unlike traditional CL approaches that focus on individual instances or tasks, our method refines the feedback loop from each training iteration to optimize the *selected batch*. This is particularly relevant in the context of "Optimization in the Wild", as it addresses the challenges posed by the increasing complexity and variety of machine learning applications.

2. Background

Batch selection and curriculum learning CL [2] and its variants like Self-Paced Learning (SPL) [12] and Hard-Example Mining (HEM) [4, 15] provide frameworks for adaptive instance selection based on difficulty or importance. These strategies have been effective in enhancing model stability and convergence but lack a universal solution. Re-weighting techniques have been explored to stabilize gradient estimates [15], with the weight metric being a pivotal element in sample selection [16]. Methods like Active Bias [4] and Recency Bias [21] achieve minimal computational overhead while focusing on prediction uncertainty as a key weight metric. Automated CL distinguishes itself by dynamically selecting tasks using algorithms like RL or bandits [6, 16]. This paper aims to build upon these foundational techniques by introducing efficient batch selection methods, particularly in the context of learning with noisy labels.

Efficient learning with noisy labels Learning with Noisy Labels (LNL) is challenging due to DNNs' propensity to memorize incorrect instances, thereby affecting model generalization [1, 25]. Strategies like multi-network and co-training methodologies [7, 24], and loss correction techniques such as reweighting and relabeling have been proposed [4, 19, 22]. Sample selection strategies like MentorNet [11] and DivideMix [13] offer alternatives by filtering out mislabeled data. While effective, these methods often introduce computational overhead. In this paper, we focus on improving batch selection methodologies to handle noisy labels more efficiently, drawing comparisons to methods like Active Bias [4].

Improving sampling efficiency by exploration Balancing exploration and exploitation is vital in batch selection for efficient training. While various strategies like ε-greedy and UCB bandits have been effective [4, 16], Boltzmann exploration has gained prominence [3, 15]. Exp3, an adversarial bandit, serves as a baseline in non-stationary environments and has shown efficacy in automated curriculum learning [6, 16]. Full reinforcement learning (RL) offers an alternative but requires multiple rollouts, diverging from our focus on sample efficiency. Our work uniquely targets batch selection, adapting Exp3 for combinatorial bandits and drawing inspiration from the Follow the Perturbed Leader (FPL) strategy. This approach distinguishes our work from prior studies focused on instance or task selection [4, 6, 16, 21]. Details on Exp3 and FPL are in the Appendix, Sec. C.

3. Methods

Combinatorial bandits for batch selection We adapt the *Follow the Perturbed Leader* (FPL) algorithm for batch selection, originally introduced by Neu and Bartók [17]. FPL operates over n rounds, maintaining a vector of weights $w_{t,i}$ for each action $\mathbf{a}_i$ in the action set $\mathcal{A}$. In our case $\mathbf{a}$ is a binary vector, such that setting an i-th action $a_i = 1$ corresponds to selecting an i-th instance $\mathbf{x}_i$. Each round perturbs these weights with noise $\boldsymbol{\rho}_t$ from distribution Q, selecting the action $\mathbf{a}_t$

that maximizes the inner product with the perturbed weight vector. While Neu and Bartók [17] used the Exp(1) distribution for Q, recent work by Honda et al. [9] suggests the Fréchet(2) distribution yields optimal regret in adversarial settings. Unlike previous implementations, we focus on reward estimation rather than loss, aligning with Exp3. This adaptation enhances performance in our context, though it may affect the original theoretical guarantees.

Algorithm 1: Follow The Perturbed Leader (Reward-guided)

```
Data: $\mathcal{A}, n, \eta, M, Q$
for $i = 1$ to $d$ do
    $w_{0,i} \leftarrow 0$ // Initialize weight vector
end
for $t = 1$ to $n$ do
    Sample $\boldsymbol{\rho}_t \sim Q$ // Sample weight perturbations
    Compute $\mathbf{a_t} = \arg\max_{\mathbf{a} \in \mathcal{A}} \langle \mathbf{a}, \eta \mathbf{w}_{t-1} + \boldsymbol{\rho}_t \rangle$ // Choose combinatorial action
    $\mathbf{r}_t \sim \nu_{a_t}$ // Draw reward vector from arm $\mathbf{a}_t$ of MAB $\nu$
    foreach $i$ with $a_{t,i} = 1$ do
        // Geometric Re-sampling
        Sample $\sigma_{t,i} \sim$ Geometric$(p_{t,i})$
        $\hat{r}_{t,i} = \min\{M, \sigma_{t,i}\} a_{t,i} r_{t,i}$ // Compute bounded reward estimate
        $w_{t,i} = w_{t-1,i} + \hat{r}_{t,i}$ // Update weight of chosen action
    end
end
```

Following action selection, for each i where $a_{t,i} = 1$, the algorithm proceeds with a geometric resampling (GR) step. Sampling from the geometric distribution estimates $1/p_i$ and in practice is not done directly, but rather by sampling arms from $Q + \eta \mathbf{w}_{t-1}$ and counting the number of iterations to re-occurrance. M is the cap on sampling size, to trade off computational efficiency with estimation accuracy. The algorithm draws a sample $\sigma_{t,i}$ from the approximated geometric distribution, and computes a bounded reward estimate $\hat{r}_{t,i}$ in the same way as Exp3, as an importance-weighted estimate, by $a_{t,i}\sigma_{t,i}r_{t,i}$.[1]

Finally, the algorithm updates the weight $w_{t,i}$ of the chosen action a_i by adding the reward estimate $\hat{r}_{t,i}$ to the previous weight $w_{t-1,i}$. This process continues for n rounds, enabling the algorithm to effectively explore and exploit the action space by balancing the current estimated rewards and the exploration noise introduced by perturbations. While FPL may require more computational resources compared to Exp3, it offers the advantage of reducing dependency on the combinatorial action space. This makes FPL a practical choice for real-world sequential decision-making tasks.

Label noise According to Song et al. [22], label noise can be either instance-independent, characterized by constant rates and probabilities, or instance-dependent, where corruption probabilities vary with data features and true labels. This study concentrates on symmetric, instance-independent noise to provide a baseline in a controlled setting.

LNL weight metric Choosing the right metric to select informative instances is still an open problem. In the field of LNL, metrics based on prediction loss [7, 11, 13, 24] and prediction un-

[1] $\sigma_{t,i}$ estimates $1/p_i$

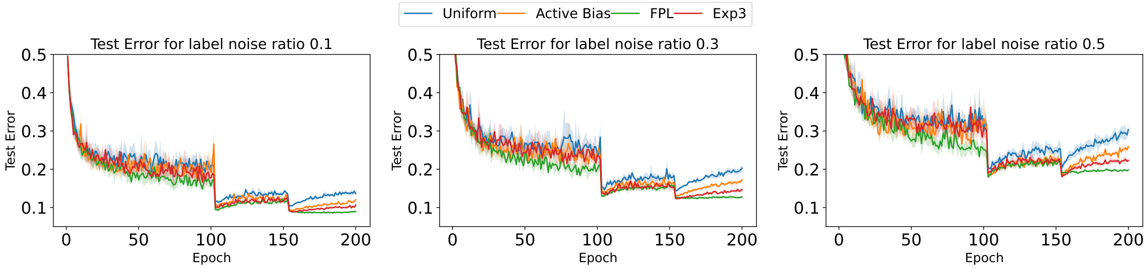


Figure 1: Test error over the course of training with confidence intervals (CI) over 5 runs for Uniform, Active Bias (weighted), Exp3 and FPL, for label noise ratio $\in \{0.1, 0.3, 0.5\}$.

certainty [4, 18, 20] have shown particular promise. The following metric was proposed by Chang et al. [4] as part of the Active Bias method:

$$w_i \propto \widehat{\text{var}}(p_{\mathcal{H}_i^{t-1}}(y_i|\mathbf{x}_i)),$$

where, for each instance $\mathbf{x}_i$, it saves prediction probabilities for their target class over time in a history buffer $\mathcal{H}_i$, and then computes their variance.

We employed this metric in our study, as it was shown suitable both for LNL and for batch selection in general. Unlike the metrics that are derived from the change in the state of the model, the probability-based metrics reflect the model's current confidence in its predictions, rendering them independent of the target solutions, and therefore, consistent across instances. This property makes them inherently balanced for problems such as LNL. However, it should be noted that while these metrics offer advantages, they do not directly track the progression of training. Therefore, following Song et al. [21], we limit the size of the history to 10 predictions.

The estimated weights serve two main purposes: either to re-weight the loss as in [19] or to parameterize the probability distribution over data instances. The latter often employs a Boltzmann distribution (e.g. [6, 16]): $P_s(i|H, S_e, D) = e^{w_i/\tau}/Z$ where Z is the normalization constant, H denotes the history of model parameters, S_e is the set of samples used in the current epoch, and $D = \{(\mathbf{x}_i, y_i) \mid i = 1, 2, \ldots, N\}$ represents the dataset. Given our interest in the role of exploration in sample efficiency, we primarily focus on sampling methods underpinned by bandit algorithms such as Exp3, which also employs a Boltzmann-like distribution.

4. Results

Experimental Setup We evaluated the performance of various sampling methods including Uniform Sampling, Active Bias, Exp3, and FPL on the CIFAR-10 dataset using a DenseNet model [10] with 40 layers. We used the Adam optimizer with momentum 0.9 and an initial learning rate of 0.1 that is decayed by multiplicative factor of 0.1 after 40 k and 60 k iterations. The batch size was set to 128 and we ran 200 epochs, consisting of 391 batches each. All methods were repeated 5 times with different seeds, under varying label corruption percentages ranging from 0% to 50%. We report the mean and 95% confidence intervals (CI) of test accuracy achieved by each method. We will release a PyTorch implementation to reproduce our experiments upon paper acceptance.

Results Under all label corruption scenarios, FPL exhibited significantly reduced noise and superior performance compared to the other methods, with Exp3 outperforming Active Bias, and Active Bias performing better than Uniform Sampling. In conditions with no label noise, no significant improvement was observed across methods, revealing a potential limitation in sensitivity to "hard to classify" instances and an overfocus on mislabeling.

Method \ Noise		10%	20%	30%	40%	50%
Uniform	best	10.24±0.25	12.05±0.54	14.08±0.39	16.75±0.30	19.40±0.26
	last	13.61±0.24	16.73±1.30	20.28±0.72	23.88±0.78	30.34±1.70
Active Bias	best	9.17±0.15	10.69±0.13	12.85±0.09	15.09±0.19	18.09±0.19
	last	12.03±0.57	13.74±0.62	17.26±0.45	20.70±0.31	25.86±0.88
Exp3	best	8.74±0.13	10.28±0.13	12.19±0.15	14.51±0.23	18.02±0.51
	last	10.64±0.22	12.30±0.49	14.67±0.80	17.42±0.51	22.27±0.50
FPL	best	**8.37±0.19**	**10.04±0.15**	**12.11±0.34**	**14.25±0.36**	**17.65±0.29**
	last	8.97±0.15	10.46±0.24	12.77±0.37	15.37±0.46	19.79±0.51

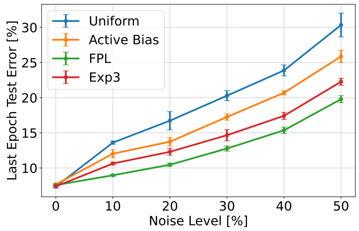


Figure 2: Left: Lowest and final epoch test errors (%) for each method on CIFAR-10 by noise ratio. Right: Visualizing last epoch performance.

Discussion The methods maintained a consistent ranking across noise levels, with the performance gap widening as noise increased (see Fig. 1 and 2). FPL consistently yielded smooth and stable convergence, due to its ability to choose informative instances. When adopting the same weight metric and neural network architecture as Active Bias, our results show that implementing a bandit strategy can lead to significant performance gains. This underscores the importance of not just selecting an optimal weight metric, but also employing a beneficial exploration policy.

Analysis of instance occurrences (Fig. 3) reveals insights into the differences in sampling strategies, addressing our initial inquiry into performance gain from utilizing batch- as opposed to instance-based feedback. Initially (Fig. 3a), all methods show similar selection frequencies, but distinctions emerge as training concludes (Fig. 3b), especially for Exp3 and FPL. Notably, the algorithm's pattern of concentration on specific instances correlates well with its performance. While Exp3 resembles an exponential distribution, FPL produces a threshold at about 20 k instances, filtering 30 k of the remaining images. The preference for 'clean' instances between the 10 k and 20 k sorted index intensifies towards the end of training, indicating the algorithm's inclination to retain instances initially deemed 'clean'. These insights emphasize FPL's efficiency as an m-set combinatorial bandit method, and highlight its suitability for batch selection.

In Fig. 3c, we display curves representing total counts throughout the run, providing a holistic view of each method's sample selection strategy. Over these, with solid lines, we superimpose the percentage of mislabeled instances within a sliding window of 1000 sorted instances. Each point on the overlay represents the mislabeling percentage within that window, revealing a trend: instances sampled less frequently (toward the right) have higher mislabeling percentages. This visualization supports our hypothesis that bandit methods with uncertainty-based metrics, like Exp3 and FPL, enhance performance by focusing on and filtering out mislabeled instances.

While Exp3 effectively identifies mislabeled instances like FPL (Fig. 3c), it tends to overfit to a narrow set and over-explore the rest (Fig. 3b). This is expected as Exp3 is an instance-based

[2] Active bias method is excluded here as we use its loss reweighting variant, and so its sampling distribution is the same as uniform.

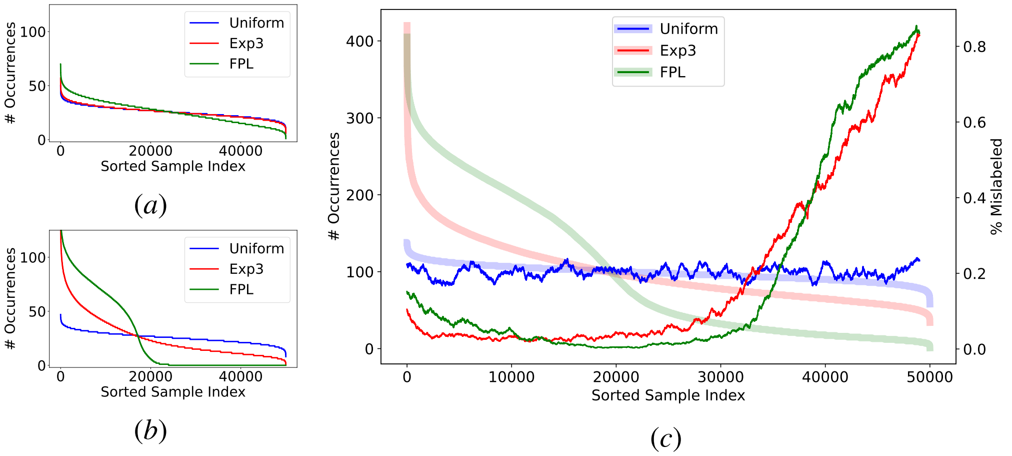


Figure 3: Analysis of instance selection for Uniform, Exp3, and FPL[2] with 20% label noise, showing selection occurrence during the initial (a) and final (b) 1000 iterations and total proportion of mislabeled instances (c; front) over total occurrences (c; background). The order of curves at index 0 aligns well with the overall performance of the methods, revealing a concentration of selection in Exp3 and FPL, particularly pronounced in FPL, with Exp3 demonstrating overfitting to a limited set of instances.

algorithm. However, this overfitting poses risks to its efficacy, as consistently selecting the same subset of impactful instances, combined with a broad array of less pertinent ones, leads to lower performance. In contrast, FPL adjusts weights to revisit a broader, balanced subset, forming more informative batches. We further analyze FPL's weight dynamics in the Appendix, Sec. A.

Scalability and hyperparameter sensitivity We ran our experiments using $\eta = 0.3$ and $\gamma = 0.1$ for Exp3 and $\eta \approx 18$, $\beta \approx 20$, and the Fréchet(0.45) distribution for FPL. To show that FPL has small sensitivity to these hyperparameters, we ran a grid search in their vicinity (see Fig. 5 in the Appendix). We found the number of GR samples to be optimal between 500 and 1000. In that range GR introduces an additional computational overhead of 20%-40%. This may seem alarming at first, however, we point out that GR is *embarrassingly parallelizable* and instance-based, which makes it scalable in practical applications.

Limitations and future directions Exp3's slower adaptation and potential benefits of its variants like Exp3.P or Exp3.IX warrant further consideration. FPL excels in balancing exploration and exploitation but shows limited improvement in noise-free scenarios, suggesting a potential overfocus on mislabeled instances. While all methods generalize well, tests were limited in scope. Future work will include naturally noisy sets like WebVision [14], as well as metrics like area under the margin (AUM) [18], to deepen insights and enhance results.

5. Conclusions

This investigation into the performance of sampling methods under different noise conditions has revealed key insights into their adaptability, stability, and algorithmic nuances. FPL's effective balance between exploration and exploitation, particularly its focus on uncertain instances, underscores its superior performance. Nonetheless, the absence of marked improvement in noise-free settings and the limited scope of our experiments highlight avenues for future research and refinement.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks, July 2017. URL http://arxiv.org/abs/1706.05394. arXiv:1706.05394 [cs, stat].

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, June 2009. Association for Computing Machinery. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

[3] Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu. Boltzmann Exploration Done Right, November 2017. URL http://arxiv.org/abs/1705.10257.

[4] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active Bias: Training More Accurate Neural Networks by Emphasizing High Variance Samples. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html.

[5] 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.

[6] Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. PMLR, 2017.

[7] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/a19744e268754fb0148b017647355b7b-Abstract.html.

[8] James Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

[9] Junya Honda, Shinji Ito, and Taira Tsuchiya. Follow-the-Perturbed-Leader Achieves Best-of-Both-Worlds for Bandit Problems. In *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, pages 726–754. PMLR, February 2023. URL https://proceedings.mlr.press/v201/honda23a.html.

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks, January 2018. URL http://arxiv.org/abs/1608.06993. arXiv:1608.06993 [cs].

[11] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels, August 2018. URL http://arxiv.org/abs/1712.05055. arXiv:1712.05055 [cs].

[12] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 1189–1197, Red Hook, NY, USA, December 2010. Curran Associates Inc.

[13] Junnan Li, Richard Socher, and Steven C. H. Hoi. DivideMix: Learning with Noisy Labels as Semi-supervised Learning, February 2020. URL http://arxiv.org/abs/2002.07394. arXiv:2002.07394 [cs].

[14] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[15] Ilya Loshchilov and Frank Hutter. Online Batch Selection for Faster Training of Neural Networks, April 2016. URL http://arxiv.org/abs/1511.06343.

[16] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE transactions on neural networks and learning systems*, 2019.

[17] Gergely Neu and Gábor Bartók. Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits. *Journal of Machine Learning Research*, 17 (154):1–21, 2016. URL http://jmlr.org/papers/v17/15-091.html.

[18] Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying Mislabeled Data using the Area Under the Margin Ranking, December 2020. URL http://arxiv.org/abs/2001.10528. arXiv:2001.10528 [cs, stat].

[19] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to Reweight Examples for Robust Deep Learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4334–4343. PMLR, July 2018. URL https://proceedings.mlr.press/v80/ren18a.html.

[20] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing Unclean Samples for Robust Deep Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5907–5915. PMLR, May 2019. URL https://proceedings.mlr.press/v97/song19b.html. ISSN: 2640-3498.

[21] Hwanjun Song, Minseok Kim, Sundong Kim, and Jae-Gil Lee. Carpe Diem, Seize the Samples Uncertain “at the Moment” for Adaptive Batch Selection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM ’20, pages 1385–1394, New York, NY, USA, October 2020. Association for Computing Machinery. ISBN 978-1-4503-6859-9. doi: 10.1145/3340531.3411898. URL https://dl.acm.org/doi/10.1145/3340531.3411898.

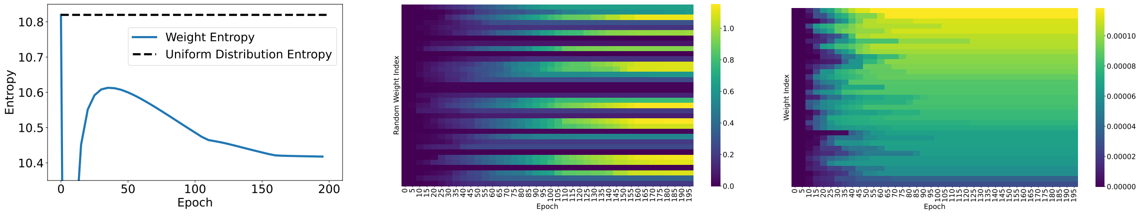
[22] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] Xin Wang, Yudong Chen, and Wenwu Zhu. A Survey on Curriculum Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, September 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3069908.

[24] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does Disagreement Help Generalization against Label Corruption?, May 2019. URL http://arxiv.org/abs/1901.04215. arXiv:1901.04215 [cs, stat].

[25] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, February 2017. URL http://arxiv.org/abs/1611.03530. arXiv:1611.03530 [cs].

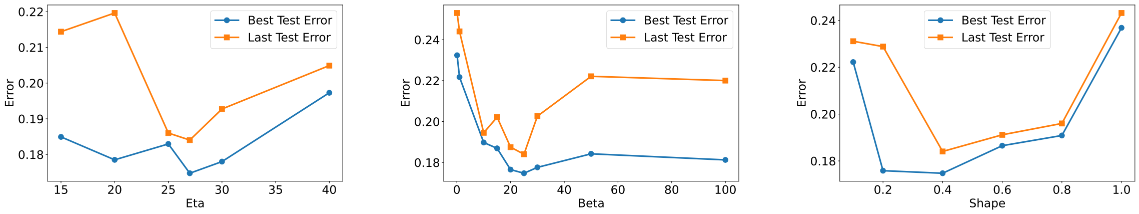
Appendix A. Analysis of weight dynamics



(*a*) Weight entropy (*b*) Weights over time (*c*) Lowest Weights sorted

Figure 4: Instance weight visualization. Entropy-aggregated weights are visualized over time in (a). Once all instances are selected by epoch 40, the entropy gradually declines as certain instances gain importance. This pattern indicates effective diversification without overfitting. In (b), a random subset of weights is displayed over time to further validate their individual trajectories. In (c) the lowest weights are shown to dynamics of weights that remain close to 0.

In this section, we provide an additional analysis of FPL’s weight dynamics as illustrated in Fig. 4. Initially set to 0, the weights adjust smoothly throughout training, maintaining a balance in instance selection without anomalies or overfitting. There is a 40%-60% split in instance selection (Fig. 3b) that is clearly reflected in the weights, with those corresponding to highly informative instances increasing rapidly, whereas those consistently labeled (either correctly or as mislabeled) remaining closer to zero. It is worth noting here that all instances were selected at least once, with all weights turning strictly positive by the end of the run. Entropy visualization (Fig. 4a) further emphasizes effective convergence on a well-sized subset of instances, reinforcing selection for better exploitation without excessive exploration.



(*a*) Error w.r.t. Eta (Beta=25.0, Shape=0.4)

(*b*) Error w.r.t. Beta (Eta=27.0, Shape=0.4)

(*c*) Error w.r.t. Shape (Eta=27.0, Beta=25.0)

Figure 5: FPL hyperparameter sensitivity analysis for 50% label noise.

Appendix B. Sensitivity analysis

The results of the sensitivity analysis of the FPL's hyperparameters described in the main body are shown in Fig. 5.

Appendix C. Bandit algorithms

C.1. Adversarial multi-armed bandit problem

A classic baseline approach for non-stationary environments is the adversarial bandit, in particular, the Exp3 algorithm and its variants. In an adversarial K-armed bandit problem, at each time step $t \in \{1, 2, ..., T\}$, the player selects an action $a_t \in \{1, 2, ..., K\}$ and then an adversary, with full knowledge of the player's previous actions, assigns a reward vector $\mathbf{r_t} = (r_{t,1}, r_{t,2}, ..., r_{t,K}) \in [0, 1]^K$ across all actions. The player receives a reward r_{t,a_t} corresponding to the selected action a_t. There is typically almost no restrictions on how the adversary can choose the reward vectors, as long as the sequence of reward vectors $\mathbf{r_1}, \mathbf{r_2}, ..., \mathbf{r_T}$ is fixed in advance or chosen based on the player's past actions. The player's goal remains to maximize the total collected reward or equivalently, to minimize regret.

The regret is typically defined with respect to the best fixed action in hindsight, i.e., the action that would have achieved the highest total reward if it had been selected at all times:

$$R(T) = \max_a \sum_{t=1}^{T} r_{t,a} - \sum_{t=1}^{T} r_{t,a_t}$$

where $R(T)$ is the total regret after T time steps, $r_{t,a}$ is the reward of action a at time t, and r_{t,a_t} is the reward of the action selected by the player at time t.

The classic adversarial algorithm, Exp3 (Alg. 2), employs a multi-step process for importance adjustments of reward estimates. First, it adjusts the reward r_j for arm j at time t using the formula

$\hat{r}_j = \frac{r_{t,j}}{p_{t,j}} \mathbb{I}_{j=i_t}$, where $p_{t,j}$ is the probability of choosing arm j. These adjusted rewards are then used to estimate sample weights w_j. Finally, these weights w_j are utilized in the Boltzmann distribution for sampling instances.

Algorithm 2: Exp3 Algorithm

```
Data: $\gamma \in (0, 1]$, $K$
for $i = 1$ to $K$ do
    $w_{1,i} \leftarrow 1$ // Initialize weights
end
for each round $t = 1, 2, ...$ do
    $p_{t,j} \leftarrow (1 - \gamma)\frac{w_{t,j}}{\sum_{k=1}^{K} w_{t,k}} + \frac{\gamma}{K}$ // Compute pmf
    $i_t \sim p_t$ // Sample action
    $r_{t,i_t} \sim \nu_{i_t}$ // Draw reward from arm $i_t$ of MAB $\nu$
    $\hat{r}_j \leftarrow \frac{r_{t,j}}{p_{t,j}} \mathbb{I}_{\{j=i_t\}}$ // Compute reward estimate
    $w_{t+1,j} \leftarrow w_{t,j} \exp(\gamma \hat{r}_{t,j})$ // Update weights
end
```

The Exp3 algorithm is very efficient computationally and is suitable for task or individual instance selection, but it doesn't take into account an impact of a full batch of instances on the performance of the neural network, which may result in suboptimal performance when competing instances are present in the same batch.

C.2. Combinatorial adversarial bandits

In order to select a full batch of instances at once we need to utilize the combinatorial bandit paradigm, which considers the joint utility of combinations of "basic arms". Formally, combinatorial bandits can be considered a type of bandit where a subset of arms is selected in a form of a binary vector $\mathbf{a} \in \{0, 1\}^d$, and the bound on regret is defined with respect to its linear combination with the reward vector:

$$R(T) = \mathbb{E}\left[\max_{\mathbf{a}} \sum_{t=1}^{T} \langle \mathbf{a_t} - \mathbf{a}, \mathbf{y_t} \rangle\right]$$

In particular, in this work we consider only the subset of a pre-specified batch size m, s.t. $||\mathbf{a}||_1 = m$.

Combinatorial bandits can be categorized as full-information, semi-bandits, or full-bandit feedback. In the full-bandit feedback scenario we observe just one reward per batch. While maximizing this reward is our ultimate objective, ignoring the available information about rewards received for individual arms makes the decision process suboptimal. The full information setup, where rewards from all the arms are observed is computationally infeasible, as it requires re-evaluating the network on all the instances. In our research we adopt the semi-bandit, in which the rewards are observed only for the basic arms selected in the current round. This setup aligns well with our problem, where we observe the rewards for instances that the neural network was trained on in current iteration. As this information is readily available, no additional passes through the network are required.

A direct application of Exp3 to the semi-bandit problem would entail monitoring the sequence of estimates for $\binom{K}{m}$ arms, a task that is computationally infeasible. The state-of-the-art approach to semi-bandits is Follow the Perturbed Leader (FPL) [8], which mimics Exp3, but estimates probabilities using reward perturbations, rather than storing them directly. Further on, Neu and Bartók [17] proposed to use geometric sampling to estimate probabilities for importance-weighted reward estimates (re-weighting step), making FPL the first computationally feasible solution to semi-bandits with strong guarantees. While other methods may offer comparable or superior theoretical performance in terms of upper regret bounds, they frequently suffer from computational inefficiency or require additional optimization steps, rendering them impractical for real-world applications. The primary appeal of FPL lies in its computational efficiency.