FOLK THEOREM-INSPIRED LOSS FUNCTION FOR ROBUST AND FAIR LEARNING UNDER CLASS IMBALANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Class imbalance biases deep classifiers toward majority classes and away from under-represented ones. We propose a loss function inspired by the Folk theorem from repeated-game theory: each class is treated, loosely, as a player whose classification accuracy is its payoff, and classes that persistently over-perform are assigned a dynamic penalty at both the batch level and the epoch level, while under-performing classes retain full gradient influence. The penalty is folded directly into each class's effective loss weight rather than added to the scalar loss value, so that it provably reshapes gradients instead of merely shifting a number (section 3). We further combine per-class gradients into a Pareto-Optimal Gradient (POG) and reconcile that direction with the overall classification gradient via a two-objective PCGrad projection. The method is loss-agnostic and is evaluated on top of SCCE, Focal, LDAM, and Class-Balanced losses, across seven CNN backbones and four benchmark datasets (CIFAR-10, CIFAR-100, Fashion-MNIST, Caltech-101) under a single, unified preprocessing and imbalance protocol. We report macro-averaged precision/recall/F1, worst-class accuracy, and class-wise F1 variance alongside accuracy, and ablate the batch-level penalty, the epoch-level penalty, and the POG+PCGrad gradient fusion independently. Evaluated across seven CNN backbones and four base losses, the proposed penalty improves accuracy and fairness metrics over the corresponding baseline loss on the large majority of backbone/dataset combinations, most consistently when combined with SCCE. All code, configuration files, and raw per-run logs are released at an anonymized repository link: https://anonymous.4open.science/r/Folk-Theorem-Loss-3D8E).

1 INTRODUCTION

Class imbalance is pervasive in real-world classification: medical diagnosis, fraud detection, and natural image recognition ,where unequal class frequencies can bias learned decision boundaries toward majority classes and degrade minority-class performance (Megahed et al., 2021; Ghosh et al., 2024). Static remedies such as oversampling, undersampling, or fixed class weights (Khan et al., 2024; Sharma et al., 2022), primarily address imbalance at the data or loss level and remain insensitive to how class-wise learning dynamics evolve during training. Sample-difficulty-based losses such as Focal loss (Lin et al., 2017) provide greater emphasis to hard examples, but difficulty and class membership can be correlated, causing minority-class samples to receive inconsistent treatment when they disproportionately constitute the hard examples. Beyond sample frequency, we identify another form of imbalance that can emerge during representation learning *feature prominence*: where the learned representation can disproportionately encode features associated with some classes even when class frequencies are balanced (Ye et al., 2020).

We propose a dynamically adaptive class reweighting loss function motivated by the Folk theorem from repeated game theory (Levin, 2006; Wikipedia contributors, 2025), which describes how repeated interaction lets a group of players sustain cooperative outcomes by threatening temporary punishment against anyone who deviates. We use this only as a design metaphor, not as a formal game-theoretic result: we treat each class as a "player" whose classification accuracy is its payoff, and we assign classes that persistently over-perform a temporary, decaying penalty at both the batch

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

level and the epoch level, so that under-performing classes retain more gradient influence until the gap narrows. We are direct about the scope of this analogy up front: supervised classification has no strategic agents, no actions, no deviations, and no credible punishment in the game-theoretic sense, so we do not claim a Nash-equilibrium proof for the training dynamics; section 6 discusses this directly. What we do claim, and evaluate empirically, is that the resulting penalty mechanism improves class-wise balance.

The penalty enters the loss as an effective per-class weight that multiplies the base cross-entropy term, so it directly scales each class's gradient contribution (section 3, Equation 6).

Our **contributions**:

- A batch-level and epoch-level dynamic class penalty, folded into each class's effective loss weight, so that the penalty provably affects the gradient (section 3).
- A gradient-fusion step combining per-class Pareto-optimal gradients (POG) with a two-objective PCGrad projection against the overall classification gradient.
- An evaluation with macro-averaged metrics, worst-class accuracy, class-wise F1 variance, a Class-Balanced loss baseline, and independent ablations of the batch penalty, the epoch penalty, and POG+PCGrad, across multiple backbones, which we found necessary for a rigorous evaluation under imbalance (section 4).

2 Related Work

Data-level and re-weighting methods. Oversampling and undersampling directly modify the training distribution (Chawla et al., 2002; Van Hulse et al., 2007; Drummond et al., 2003) but respectively risk overfitting to synthetic samples or discarding information. Cost-sensitive re-weighting instead scales each sample's loss, typically by inverse class frequency (Cui et al., 2019; Huang et al., 2016) or its square root (Reed et al., 2015), tracing back to importance sampling (Owen, 2013). Difficulty-based re-weighting (Jiang et al., 2018; Katharopoulos & Fleuret, 2018) and Focal loss (Lin et al., 2017) instead down-weight easy examples, but in imbalanced data this can inadvertently suppress minority classes, whose examples are often also the hard ones; LDAM (Cao et al., 2019) instead regularizes per-class margins to counter this. Two-stage fine-tuning decouples representation learning from classifier re-balancing (Kang et al., 2019; Menon et al., 2021; Zhang et al., 2021b;a). None of these methods adapts *during* training to the model's currently observed class-wise performance, which is the gap we target.

Meta-learning. Such approaches (Finn et al., 2017; Snell et al., 2017; Santoro et al., 2016) address few-shot and imbalanced settings by learning to adapt quickly to new task or low-data regimes. This is complementary to our objective, which dynamically adjusts class contributions within supervised training rather than learning a task-adaptation procedure.

Gradient-conflict resolution. Multi-task and multi-objective optimization methods address conflicting gradients using Pareto-based optimization and projected gradient surgery. Our POG+PCGrad formulation extends gradient conflict resolution from task-level to class-level gradients, explicitly accounting for conflicts among class-specific learning signals to our POG+PCGrad step. (Extended discussion is in subsection A.5.)

3 Method

3.1 Preliminaries

Let $\{(x_i, y_i)\}_{i=1}^{N}$ be a training batch with C classes, $z_{i,j}$ the logit for class j on sample i, and $y_i \in \{1, \ldots, C\}$ the true label. The base loss is sparse categorical cross-entropy,

$$\mathcal{L}_{\text{ce},i} = -\left(z_{i,y_i} - \log \sum_{j=1}^{C} e^{z_{i,j}} \right). \tag{1}$$

Every class c starts from a static weight w_c; unless stated otherwise (as for the class-balanced baseline, which uses its own effective-number weights (Cui et al., 2019)) w_c is the standard inverse-

frequency balanced weight,

$$w_c = \frac{N_{\text{train}}}{C \cdot n_c}, \tag{2}$$

where n_c is the number of training samples of class c. We state this explicitly since w_c is not learned and is fixed for the whole run, before any Folk penalty is applied on top of it.

Per-batch class-wise accuracy, used by both penalty terms below, is

$$\text{ACC}_c = \frac{\sum_{i=1}^{N} \mathbb{1}(y_i = c)\,\mathbb{1}(\arg\max_j z_{i,j} = c)}{\sum_{i=1}^{N} \mathbb{1}(y_i = c)}. \tag{3}$$

ACC_c is a non-differentiable function of the model's predictions (it involves an $\arg\max$). ACC_c is never backpropagated through. It is computed out of gradient calculations and used only to update the penalty state P_c, F_c below, which are themselves treated as constants at the point they multiply the loss. No differentiability is required or claimed for ACC_c itself.

3.2 Batch-level penalty P_c

$$P_c \leftarrow \begin{cases} P_c + \lambda, & \text{if } \text{ACC}_c < \text{ACC}_c^{\text{prev}} \\ \max(P_c - \lambda,\ 0), & \text{if } \text{ACC}_c > \text{ACC}_c^{\text{prev}} \end{cases} \tag{4}$$

where $\text{ACC}_c^{\text{prev}}$ is class c's accuracy the last time it appeared in a batch, and $\lambda = 0.001$. P_c is floored at 0 by construction (Equation 4) but has no explicit ceiling: in practice it cannot grow without bound, because once a class's accuracy stops declining P_c immediately starts decaying at the same rate λ it grew; whether this is enough is an empirical question we ablate directly (section 4), rather than something we prove analytically. We deliberately did not add a hard ceiling so that the ablation reflects the mechanism as originally proposed.

3.3 Epoch-level Folk penalty F_c

At the end of each epoch, let $a = \min_c \text{ACC}_c$ and $b = \max_c \text{ACC}_c$ over that epoch's class-wise accuracies. The Folk penalty is

$$F_c = \begin{cases} 0, & a = b \\ \dfrac{(b-a)\lambda_f}{e^b - e^a}\, e^{\text{ACC}_c} \;-\; \dfrac{(b-a)\lambda_f}{e^b - e^a}\, e^a, & a \neq b \end{cases} \tag{5}$$

with $\lambda_f = 0.01$. This is a re-parameterization of e^{ACC_c} that passes through $(a, 0)$ and (b, λ_f), so the lowest-accuracy class of the epoch gets zero penalty and the highest-accuracy class gets at most λ_f; the full derivation, including the removable-singularity limit used when $a \to b$, is in subsection A.1. Note that F_c is monotonically increasing in ACC_c by construction: higher-accuracy classes are penalized more, up to the λ_f cap, which is the intended direction of the mechanism.

3.4 Effective weight

The batch- and epoch-level penalties are folded directly into each class's loss weight, rather than into the loss value:

$$w_c^{\text{eff}} = \max\big(w_c + P_c + F_c,\ 0\big), \qquad \mathcal{L}_{\text{final}} = \frac{1}{N} \sum_{i=1}^{N} w_c^{\text{eff}}\, \mathcal{L}_{\text{ce},i}. \tag{6}$$

Because w_c^{eff} multiplies the base cross-entropy term, $\partial \mathcal{L}_{\text{final}} / \partial \theta = \frac{1}{N} \sum_i w_c^{\text{eff}}\, \partial \mathcal{L}_{\text{ce},i} / \partial \theta$: the penalty scales each sample's gradient contribution directly, which is what lets it strengthen or weaken a class's influence on training. The floor at 0 keeps a majority class's weight from going negative (which would reverse its gradient) even under severe imbalance; combined with P_c, F_c both being bounded ($F_c \leq \lambda_f$, and P_c's practical bound as discussed above), a majority class's effective weight is reduced, not eliminated; we return to whether this reduction is large enough in section 6. Equation 6 is architecture- and base-loss-agnostic: $\mathcal{L}_{\text{ce},i}$ can be swapped for Focal, LDAM, or Class-Balanced per-sample losses without changing anything else (section 4).

3.5 GRADIENT FUSION: POG + PCGRAD

Let $g_{\text{loss}} = \nabla_\theta \mathcal{L}_{\text{final}}$ and, for each class c present in the batch, $g_c = \nabla_\theta \mathcal{L}_c$ where $\mathcal{L}_c$ is the mean effective loss of class c's samples in the batch. We combine the $\{g_c\}$ into a single Pareto-optimal direction $g_{\text{pog}} = \sum_c x_c g_c$ by solving for non-negative, simplex-constrained weights x_c that minimize $\|\sum_c x_c g_c\|$ on the Gram matrix $G = [g_c^\top g_{c'}]$ (regularized for numerical stability), following Pareto Optimal Gradient combination. We then resolve any conflict between g_{pog} and the overall gradient g_{loss} with a two-objective PCGrad projection:

$$g'_{\text{loss}} = \begin{cases} g_{\text{loss}} - \dfrac{\langle g_{\text{loss}}, g_{\text{pog}} \rangle}{\|g_{\text{pog}}\|^2 + \epsilon} g_{\text{pog}}, & \langle g_{\text{loss}}, g_{\text{pog}} \rangle < 0 \\ g_{\text{loss}}, & \text{otherwise} \end{cases}, \qquad g_{\text{final}} = \tfrac{1}{2}\big(g'_{\text{loss}} + g_{\text{pog}}\big). \tag{7}$$

POG runs first because it must reconcile up to C mutually competing directions; PCGrad runs second because it is a two-objective operator, so it is only well-posed once POG has already collapsed the per-class gradients into one direction. We ablate POG+PCGrad independently from the two penalties (section 4) and report its wall-clock cost directly, since it requires one additional backward pass per class present in a batch (Table 4).

4 EXPERIMENTAL SETUP

We evaluate whether the proposed method improves class-balanced predictive performance under controlled class imbalance. All methods are trained using the same data splits, preprocessing, architectures, optimization budgets, and model-selection criterion. Unless stated otherwise, only the loss and gradient-combination procedure vary between experiments.

Datasets. We use four datasets: CIFAR-10, CIFAR-100, Fashion-MNIST (Krizhevsky et al., 2025; Xiao et al., 2017), and Caltech-101 (Li et al., 2022). CIFAR-10/100 and Fashion-MNIST are artificially imbalanced; Caltech-101 is naturally imbalanced (40–800 images per class). Every dataset – including Caltech-101 – is resized to 75×75 and goes through the *same* pipeline, so there is exactly one resize path shared by all four datasets (subsection A.3). Splits are 7:1:2 train/val/test, with the Buda et al. (Buda et al., 2018) step imbalance protocol applied to the train split only: for imbalance factor α, even-indexed classes retain $\lfloor \alpha n_c \rfloor$ samples and odd-indexed classes retain all n_c samples.

Backbones. Seven CNNs, matching the original study: Xception, MobileNetV2, ResNet50, DenseNet201, InceptionV3, EfficientNetB0, and a NASNetMobile substitute (MnasNet), a similarly sized (∼4–5M parameter) mobile, NAS-derived backbone.

Base losses and baselines. SCCE, Focal (Lin et al., 2017), LDAM (Cao et al., 2019), and Class-Balanced (CB) loss (Cui et al., 2019). CB is included as a baseline since it is a natural comparison point for the effective-weight formulation in Equation 6. Each base loss is run both with and without the Folk penalty (Equation 6).

Metrics. Accuracy alone is not an appropriate primary metric under class imbalance, since it is dominated by majority-class performance. We report macro-averaged precision, recall, and F1 throughout, plus two fairness-oriented metrics: worst-class accuracy and class-wise F1 standard deviation. All metrics are computed identically, in one shared evaluation function, for every (dataset, backbone, loss) combination.

Ablations. We isolate the Folk penalty and POG+PCGrad by running four variants: plain, Folk-only, POG+PCGrad-only, and the full mechanism; on MobileNetV2 (Table 3). We also report the per-epoch wall-clock overhead of POG+PCGrad relative to a plain weighted-loss backward pass (Table 4), since it requires one extra backward pass per class present in a batch and its cost therefore scales with C.

Implementation. All experiments use a single PyTorch codebase: one data pipeline, one model registry, one loss module, and one training loop, run sequentially over every (dataset, backbone, loss, ablation-variant) combination, each writing checkpoints, logs, and metrics to its own directory. subsection A.3 has the code-availability link.

5 RESULTS

Across most backbone/dataset/loss combinations, adding the Folk penalty improves or maintains accuracy and F1 relative to the corresponding base loss, with the most consistent gains observed when combined with SCCE (Table 1, Table 6). Table 2 gives a fairness metrics on CIFAR-10 using Xception modelwith SCCE base loss. The gains are not uniform: some backbone/loss/dataset combinations show only modest improvements. The ablations in Table 3 and Table 8 further show the contributions of the individual penalty components and POG+PCGrad. We have not yet evaluated multiple random seeds; therefore, differences on the order of 0.5–1% should not be interpreted as statistically significant without further evidence. Grad-CAM, activation-map, and t-SNE visualizations, together with the full literature-baseline comparison, are provided in subsection A.9 and subsection A.7, respectively.

Table 1: Results on CIFAR-100 across different backbones. We report accuracy and F1 for the baseline SCCE loss and SCCE with the proposed Folk penalty. Best result within each pair is shown in bold. All results are in %

	Accuracy		F1	
Backbone	Plain	+Folk	Plain	+Folk
Xception	64.31	**70.71**	48.82	48.82
MobileNetV2	44.58	**60.83**	44.58	**60.83**
ResNet50	65.79	**66.01**	44.33	**47.16**
DenseNet201	65.23	**68.41**	45.49	**51.04**
InceptionV3	65.91	**68.34**	46.16	**51.78**
NASNetMobile	63.82	**65.78**	44.09	**55.68**
EfficientNetB0	63.91	**68.44**	44.48	**46.83**

Table 2: Fairness metrics on CIFAR-10 using Xception with SCCE base loss.

Metric	Plain	+Folk
Worst-class accuracy	72.40	75.33
Class-wise F1 std. dev.	5.79	6.16

Table 3: Ablation of the proposed components on CIFAR-100 using MobileNetV2 with respect to SCCE.

Backbone	Plain	Folk only	POG/PCGrad only	Full (proposed)
MobileNetV2	44.58	60.82	60.34	60.83

Table 4: Per-epoch training wall-clock (seconds), plain vs. POG+PCGrad, same backbone and batch size.

Dataset (C classes)	Plain	+POG/PCGrad
Fashion-MNIST (C=10)	4.6 s	23.8 s
CIFAR-100 (C=100)	3.9 s	72.9 s

6 LIMITATIONS

The Folk-theorem framing is an analogy, not a proof. Supervised classification does not involve autonomous agents, strategic actions, or credible punishment in the game-theoretic sense. We therefore do not claim that the proposed optimization dynamics constitute a repeated game or converge

to a Nash equilibrium. Characterizing the stability or convergence of the resulting stochastic, non-convex optimization process remains an open theoretical problem.

Majority-class weight reduction. Equation 6 floors w_c^{eff} at 0 rather than letting it go negative, but a severely imbalanced majority class can still see its effective weight reduced substantially, which could in principle degrade the overall accuracy on that class. We have not broken results down by majority-class per-class accuracy specifically; doing so directly, rather than relying on the aggregate metrics in section 5, would confirm whether this concern materializes in practice.

Datasets and backbones are conventional. Our evaluation considers four image-classification datasets and conventional CNN backbones under a step-imbalance setting and do not extend to tansformer architectures, which limits how much this evaluation says about long-tailed ImageNet-scale or iNaturalist-scale settings. We did not add such a dataset in this version due to compute budget; this is future work, not a currently-addressed gap.

POG+PCGrad cost scales with the number of classes. The POG+PCGrad component requires class-specific gradients for the classes represented in each mini-batch (Table 4). This can limit scalability to large-class problems and large architectures. The additional computation is incurred only during training and does not introduce inference-time overhead. More extensive measurements of wall-clock time, memory usage, and scaling with the number of classes would clarify the regimes in which gradient fusion provides sufficient benefit relative to penalty-only training.

7 Conclusion

In this work, we identified class-wise learning imbalance as a source of performance disparity in imbalanced classification, arising not only from unequal class frequencies but also from differences in feature prominence and learning dynamics. We proposed a dynamically adaptive loss that adjusts the influence of each class according to its observed performance during training, using batch-level and epoch-level penalties.While our formulation is motivated by the Folk theorem, we use the game-theoretic interpretation as a design perspective rather than a formal characterization allowing underperforming classes to retain greater influence during optimization. We further introduced POG+PCGrad to account for conflicts among class-specific gradients. Across multiple benchmarks and CNN backbones, our approach improves class-balanced performance while reducing disparities in class-wise learning. Through controlled ablations, we show the complementary contributions of the adaptive penalties and gradient-fusion component providing evidence that both adaptive reweighting and gradient fusion contribute to the observed improvements. We hope this work motivates further investigation of adaptive, performance aware optimization strategies for imbalanced learning.

Reproducibility Statement

The full PyTorch implementation – data/imbalance pipeline, model registry, loss module (including Equation 6 and the POG+PCGrad step), training loop, and SLURM launch scripts – will be released at the anonymized link in subsection A.3, together with every run's per-epoch metrics CSV, final test-set predictions, and configuration JSON, so every number in section 5 traces back to a specific run directory. Hyperparameters are listed in subsection A.4.

Ethics Statement

This work uses only public benchmark image datasets (CIFAR-10, CIFAR-100, Fashion-MNIST, Caltech-101) and does not involve human subjects, personal data, or any dataset release.

References

Shakhnaz Akhmedova and Nicolas Körber. Next generation loss function for image classification. In *International Conference on Pattern Recognition*, pp. 334–349. Springer, 2024.

Alessia Amelio, Gianluca Bonifazi, Enrico Corradini, Domenico Ursino, and Luca Virgili. A multilayer network-based approach to represent, explore and handle convolutional neural networks. *Cognitive Computation*, 15:61–89, 2022.

Tianhao Bu, Michalis Lazarou, and Tania Stathaki. Image edge enhancement for effective image classification. *arXiv preprint arXiv:2401.07028*, 2024.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32, 2019.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Zhi Chen, Jiang Duan, Li Kang, and Guoping Qiu. Class-imbalanced deep learning via a class-balanced ensemble. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5626–5640, 2021.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Chris Drummond, Robert C Holte, et al. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pp. 1–8, 2003.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Kushankur Ghosh, Colin Bellinger, Roberto Corizzo, Paula Branco, Bartosz Krawczyk, and Nathalie Japkowicz. The class imbalance problem in deep learning. *Machine Learning*, 113 (7):4845–4901, 2024.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384, 2016.

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. In *International Conference on Machine Learning (ICML)*, pp. 2304–2313, 2018.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Advances in Neural Information Processing Systems*, 2019.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine Learning (ICML)*, pp. 2525–2534, 2018.

Alif Elham Khan, Mohammad Junayed Hasan, Humaira Anjum, and Nabeel Mohammed. Shadow: A novel loss function for efficient training in siamese networks. *arXiv preprint arXiv:2311.14012*, 2023.

Azal Ahmad Khan, Omkar Chaudhari, and Rohitash Chandra. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Systems with Applications*, 244:122778, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html, 2025. Accessed 08-02-2025.

Jonathan Levin. Repeated games i: Perfect monitoring. Lecture notes, Stanford University, 2006.

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101. https://doi.org/10.22002/D1.20086, 2022.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

Sunwei Liu, Xin Li, Xiang Liang, Rong Jin, and Liang Lin. Long-tail learning without tuning. In *International Conference on Learning Representations*, 2022.

Fadel M Megahed, Ying-Ju Chen, Aly Megahed, Yuya Ong, Naomi Altman, and Martin Krzywinski. The class imbalance problem. *Nature Methods*, 18(11):1270–1272, 2021.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Art B Owen. *Monte Carlo theory, methods and examples*. Stanford University, 2013.

Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 735–744, 2021.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *International Conference on Learning Representations (ICLR)*, 2015.

Bo Ren, Jing Feng, and Ling Zhou. A novel loss based on stepwise gradient penalty for multi-class imbalanced data classification. *Journal of Classification*, 2025. https://doi.org/10.1007/s00357-025-09517-2.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

Shivani Sharma, Anjana Gosain, and Saroj Jain. A review of the oversampling techniques in class imbalance problem. In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 1*, pp. 459–472. Springer, 2022.

Raunak Singhal, Arnav Biswas, Sundar Elangovan, Sagar Sabnis, and Udayan Ganguly. N multipliers for n bits: Learning bit multipliers for non-uniform quantization. In *NeurIPS 2024 Workshop Machine Learning with new Compute Paradigms*, 2024.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jalaledin Toosifar, Yahya Forghani, Seyed Amin Hosseini, and Nastaran Shoeibi. A novel sub-network level ensemble deep neural network with a regularized loss function to improve prediction performance. *Journal of Big Data*, 12(1):155, 2025.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 935–942, 2007.

Wikipedia contributors. Folk theorem (game theory) — wikipedia. https://en.wikipedia.org/wiki/Folk_theorem_(game_theory), 2025. Accessed 15-03-2025.

Yong Xian, Hong Yu, Yun Wang, and Guoyin Wang. Exploring multi-granularity balance strategy for class incremental learning via three-way granular computing. *Brain Informatics*, 12(1):7, 2025.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*, 2020.

Haohan Zhang, Jindong He, Zeyi S Wang, Zhiting Wang, and Alex Lee. Understanding the generalization benefit of model invariance from a data augmentation perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a.

Yifan Zhang, Tan Wang, Guosong Long, Chao Shen, and Chunhua Zhang. Distribution alignment: A unified framework for long-tail visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, 31, 2018.

A Appendix

A.1 Derivation of the Folk penalty curve

We want a function of ACC_c that passes through $(a, 0)$ and (b, λ_f), where $a = \min_c \mathrm{ACC}_c$ and $b = \max_c \mathrm{ACC}_c$ for the epoch. Starting from $f(x) = e^x$ and re-parameterizing $g(x) = Af(x) + B$ so that $g(a) = 0$ and $g(b) = \lambda_f$ gives $A = \lambda_f/(e^b - e^a)$ and $B = -\lambda_f e^a/(e^b - e^a)$, hence Equation 5. When $a \to b$ (all classes equally accurate), write $b = a + \Delta$:

$$\frac{b-a}{e^b - e^a} = \frac{\Delta}{e^a(e^\Delta - 1)} \xrightarrow{\Delta \to 0} e^{-a},$$

using $\lim_{\Delta \to 0} \Delta/(e^\Delta - 1) = 1$, so $F_c \to \lambda_f \left(e^{\mathrm{ACC}_c - a} - 1\right)$, which is exactly 0 when $\mathrm{ACC}_c = a = b$: the penalty vanishes with no discontinuity when every class is equally accurate, and our implementation uses this closed form directly whenever $b - a$ is numerically indistinguishable from zero, rather than dividing by $e^b - e^a \approx 0$.

A.2 Algorithm listing

Algorithm 1 One training step with the Folk loss (effective-weight formulation)

Compute logits z_i for the batch; compute $\mathcal{L}_{\mathrm{ce},i}$ (Equation 1)
$w_c^{\mathrm{eff}} \leftarrow \max(w_c + P_c + F_c, 0)$ ▷ P_c, F_c detached, no grad
$\mathcal{L}_{\mathrm{final}} \leftarrow \frac{1}{N} \sum_i w_c^{\mathrm{eff}} \mathcal{L}_{\mathrm{ce},i}$ ▷ Equation 6: multiplies, does not add
if POG+PCGrad enabled **then**
    $g_{\mathrm{loss}} \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{final}}$; $g_c \leftarrow \nabla_\theta \mathcal{L}_c$ for each class c in batch
    $g_{\mathrm{pog}} \leftarrow$ Pareto-optimal combination of $\{g_c\}$
    $g_{\mathrm{final}} \leftarrow \mathrm{PCGrad}(g_{\mathrm{loss}}, g_{\mathrm{pog}})$ (Equation 7)
    $\theta \leftarrow \theta - \eta\, g_{\mathrm{final}}$
else
    $\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}_{\mathrm{final}}$
end if
no_grad: update P_c (Equation 4) from this batch's ACC_c; accumulate epoch-level correct/total counts
at epoch end, no_grad: update F_c (Equation 5) from the epoch's accumulated ACC_c

A.3 Code availability

Code, configuration files, and training logs will be made publicly available at https://anonymous.4open.science/r/Folk-Theorem-Loss-3D8E (kept anonymized for now, per double-blind review).

A.4 Hyperparameters

Table 5: Hyperparameters used across all runs unless noted otherwise.

Optimizer	Adam (Kingma & Ba, 2014)
Learning rate	0.001
Image size	75×75 (all datasets)
Train/val/test split	7:1:2
Imbalance factor α	0.1
Batch-level penalty step λ	0.001
Epoch-level penalty cap λ_f	0.01
Epochs	150
Batch size	64

λ and λ_f were chosen analytically (subsection A.1) rather than by grid search, and we did not retune them per architecture or dataset. We have not run a formal sensitivity sweep over these two values; doing so is worth adding if compute allows.

A.5 Extended related work

Data-level approaches mitigate class imbalance by directly manipulating the training distribution through re-sampling. Over-sampling introduces repeated or synthetic samples for minority classes (Chawla et al., 2002; Van Hulse et al., 2007), but this can lead to overfitting on noise in the generated samples; interpolation-based or synthetic generation techniques remain error-prone for the same reason. Under-sampling is often preferred over over-sampling (Drummond et al., 2003) since it avoids that overfitting risk, at the cost of discarding potentially valuable samples – a cost that grows with the severity of the imbalance.

Re-weighting approaches adjust sample importance directly, typically inversely proportional to class frequency (Cui et al., 2019; Huang et al., 2016) or its square root (Reed et al., 2015), tracing back to importance sampling in statistics (Owen, 2013). Cui et al. (Cui et al., 2019) proposed using the effective number of samples as a theoretically grounded re-weighting scheme (the basis of the Class-Balanced baseline in section 4); such methods, however, weight all samples within a class uniformly, regardless of individual sample difficulty. A second line of work instead re-weights by sample difficulty or loss (Jiang et al., 2018; Katharopoulos & Fleuret, 2018), down-weighting well-classified samples in favor of hard examples. Focal loss (Lin et al., 2017) is the best-known instance, but in imbalanced data hard examples often come disproportionately from majority classes, so difficulty-based weighting can inadvertently suppress minority-class gradients further; overfitting to hard examples is also a standing concern given DNN capacity (Zhang et al., 2021a). LDAM (Cao et al., 2019) instead regularizes per-class margins directly to prevent this kind of overfitting on minority classes.

Two-stage fine-tuning separates representation learning from classifier re-balancing: a first stage trains on the imbalanced data with plain cross-entropy to learn generalizable representations, and a second stage freezes the backbone and fine-tunes the classifier with class-balanced re-sampling (Kang et al., 2019), logit adjustment (Menon et al., 2021), or distribution calibration (Zhang et al., 2021b). More recent work shows representation bias can itself be significant (Liu et al., 2022), challenging the assumption that classifier bias alone dominates, and motivating joint or dynamic re-balancing across both.

Meta-learning addresses imbalance and few-shot settings by learning to adapt quickly to new class distributions: optimization-based methods such as MAML (Finn et al., 2017), metric-based methods such as Prototypical Networks (Snell et al., 2017), and memory-augmented networks (Santoro et al., 2016) all enable rapid adaptation with limited per-class examples. This is a complementary rather than competing direction to the dynamic, in-training reweighting studied here.

A.6 Method illustrations

This section gives a complete pictorial description of our approach. Figure 1 provides a full description of our pipeline. Figure 2 highlights how the folk theorem makes better clustering of features, reducing outliers in feature maps. Figure 3 provides the folk penalty curve used to apply the F_c penalty showing three cases on how the curve changes dynamically. The full description of the working and idea of POG + PCGrad has been potrayed in Figure 4

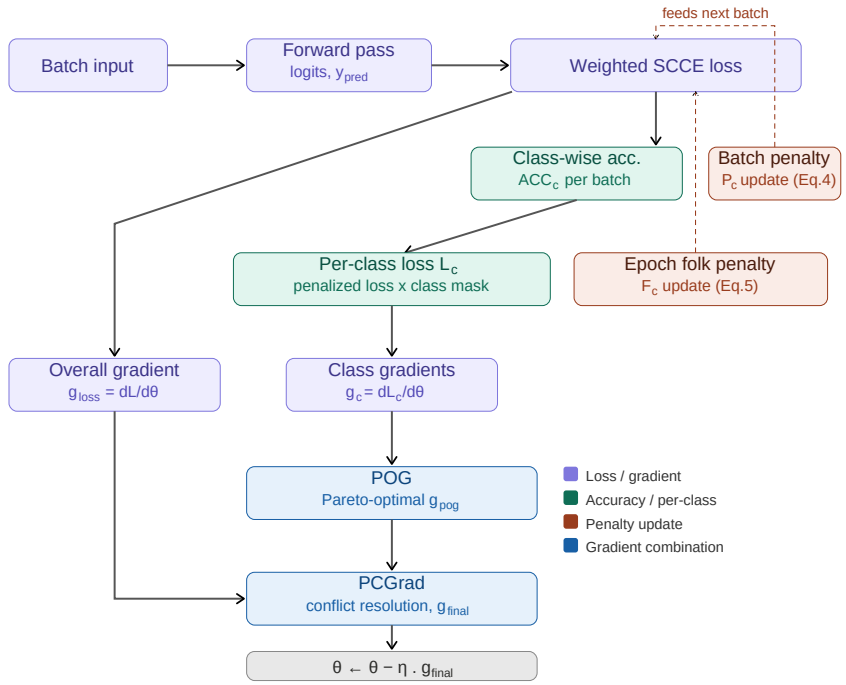

Figure 1: Overview of the training pipeline. At each batch, class-wise accuracies are computed and used to update the batch-level penalty P_c (Equation 4). At the end of each epoch, the Folk penalty F_c is updated from the global accuracy spread (Equation 5). Per-class losses incorporating both penalties are used to compute class-wise gradients g_c, combined via Pareto Optimal Gradient (POG) into g_{pog}; PCGrad then resolves conflicts between g_{pog} and the overall loss gradient g_{loss}, yielding the final update g_{final}.

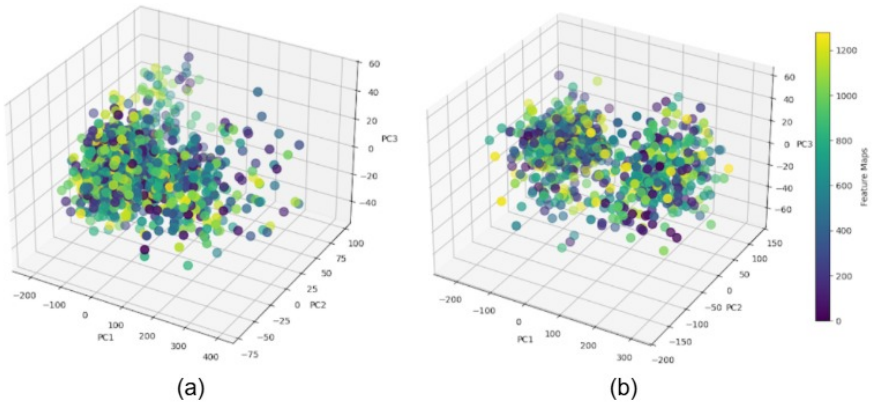

Figure 2: 3-D PCA embeddings of feature-map activations for (a) MobileNetV2 + Focal and (b) MobileNetV2 + Focal + Folk. Adding the Folk penalty produces a more spherical cluster, reduces outliers, and spreads variance more isotropically across PC1–PC3.

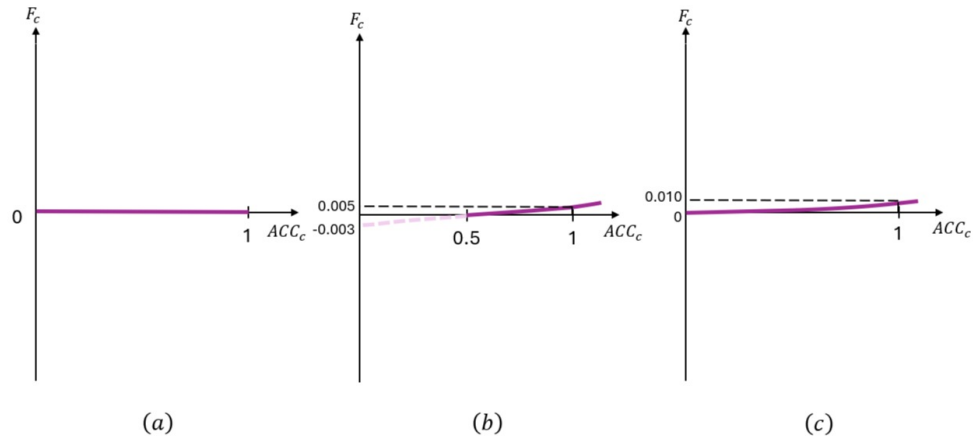

Figure 3: The Folk penalty curve F_c (Equation 5): (a) the theoretical case where all class accuracies are equal ($F_c \equiv 0$), (b) minimum class accuracy 0.5 and maximum 1, and (c) the theoretical extreme of minimum accuracy 0 and maximum 1, where the highest-accuracy class reaches the λ_f cap.

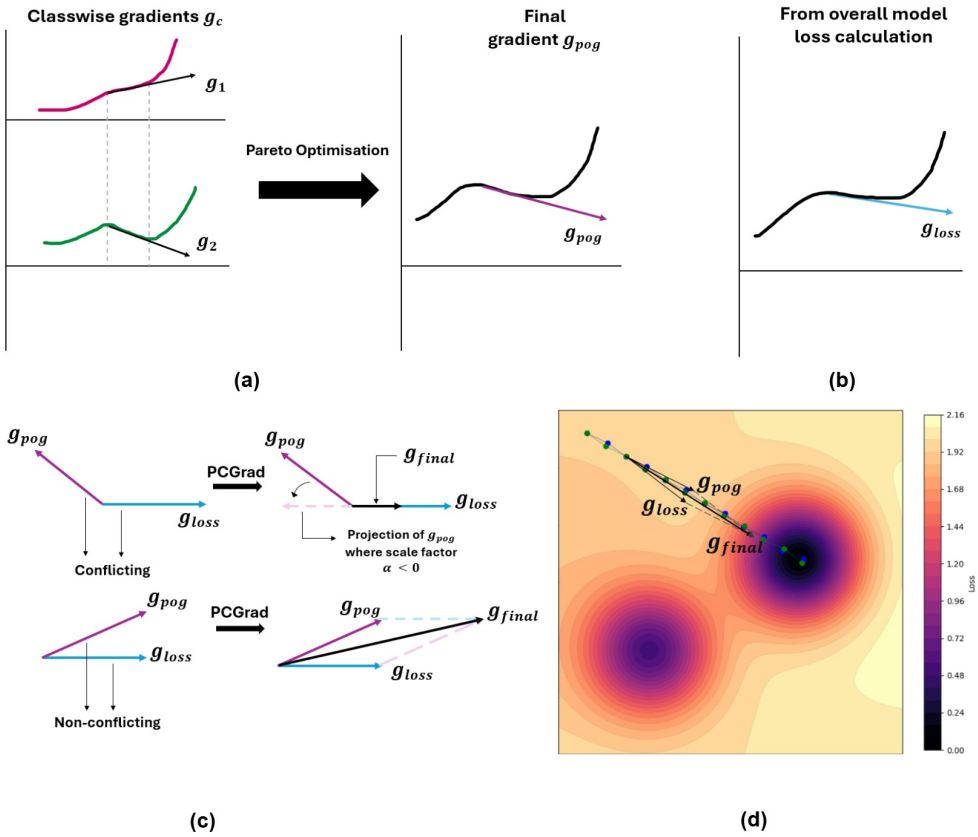

Figure 4: (a) Class-wise gradients combined into the Pareto-optimal gradient g_{pog}. (b) The gradient from the overall (weighted) loss, g_{loss}. (c) How PCGrad resolves conflicting vs. non-conflicting cases between g_{pog} and g_{loss}. (d) A 2-D loss landscape contour plot showing the trajectories of g_{loss}, g_{pog}, and the resulting g_{final}.

Table 6: Accuracy / Precision / Recall / F1 for each model on all four datasets, with SCCE, Focal, LDAM, and CB losses, and with each combined with the proposed Folk penalty. All values in %.

Losses	Models	CIFAR-10				CIFAR-100				Fashion-MNIST				Caltech-101			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
SCCE	Xception	90.67	91.02	90.67	90.72	64.31	48.67	48.91	48.82	94.46	94.46	94.46	94.44	80.12	78.54	79.24	82.24
	MobileNetV2	88.12	88.25	88.12	88.14	44.58	42.20	44.42	44.58	93.99	93.99	93.99	93.99	81.68	83.95	81.68	81.73
	ResNet50	84.87	85.05	84.87	84.86	65.79	47.21	45.67	44.33	93.93	93.93	93.93	93.84	70.37	71.99	70.37	70.12
	DenseNet201	88.87	88.95	88.87	88.84	65.23	45.74	45.21	45.49	93.40	93.44	93.40	93.41	79.50	80.96	79.50	79.03
	InceptionV3	84.04	84.08	84.04	84.01	65.91	46.11	46.21	46.16	94.05	94.05	94.05	94.05	72.78	69.25	72.24	73.45
	NASNetMobile	85.44	86.13	85.44	85.40	63.82	44.21	43.98	44.09	93.37	93.41	93.37	93.38	62.38	65.06	62.38	62.21
	EfficientNetB0	89.98	90.14	89.59	88.93	63.91	44.53	44.41	44.48	94.38	94.40	94.31	93.80	80.02	75.45	80.12	82.24
Focal Loss	Xception	81.83	66.49	65.09	64.01	55.11	24.81	25.48	24.26	90.62	90.32	90.09	89.43	72.50	77.99	72.50	73.31
	MobileNetV2	79.51	64.89	64.50	62.75	51.38	23.28	23.66	22.59	86.57	86.50	86.18	85.34	79.30	75.40	80.24	78.90
	ResNet50	70.23	72.22	70.23	69.18	50.82	25.80	26.71	25.33	88.77	88.43	88.39	87.39	54.51	60.56	54.51	55.10
	DenseNet201	66.82	76.72	55.35	59.24	55.73	23.57	23.71	23.69	88.52	88.06	88.04	87.07	70.20	74.54	70.20	70.80
	InceptionV3	77.78	79.71	76.95	75.60	56.41	24.21	24.71	24.36	83.67	83.08	83.42	82.05	76.24	72.18	75.45	78.25
	NASNetMobile	79.53	64.68	63.43	62.11	54.32	22.31	22.48	22.29	85.34	85.48	85.16	83.20	64.39	74.31	64.39	66.42
	EfficientNetB0	70.46	65.34	70.32	68.56	54.41	22.63	22.98	22.56	84.45	84.97	84.29	83.06	79.39	81.84	79.34	81.84
LDAM Loss	Xception	82.40	83.35	82.40	82.40	63.77	65.19	63.77	63.84	94.39	94.39	94.39	94.36	71.79	74.35	71.79	72.05
	MobileNetV2	78.19	80.30	78.19	78.28	60.17	62.53	60.17	59.98	93.86	93.86	93.86	93.85	75.78	79.70	75.78	76.08
	ResNet50	60.52	61.66	60.52	60.56	54.13	55.09	54.13	53.68	93.35	93.33	93.35	93.32	72.01	75.25	72.01	71.67
	DenseNet201	53.83	63.63	33.35	40.96	54.63	56.10	54.63	54.75	93.63	93.68	93.63	93.62	65.23	67.82	65.23	65.01
	InceptionV3	50.66	51.21	50.66	50.55	50.95	52.48	50.95	51.70	93.09	93.10	93.09	93.09	68.12	66.45	64.48	65.89
	NASNetMobile	64.39	74.22	64.39	63.42	57.27	63.42	57.27	57.66	93.44	93.48	93.44	93.45	61.02	66.85	61.02	61.89
	EfficientNetB0	68.76	79.26	68.76	69.67	51.38	50.63	51.38	51.01	93.54	93.65	93.54	93.58	85.78	87.31	85.78	85.70
CB Loss	Xception	86.35	87.11	86.39	86.41	54.31	59.28	54.46	52.81	92.05	92.15	92.06	92.02	66.49	59.70	53.56	49.32
	MobileNetV2	84.38	85.60	84.41	84.53	54.17	56.63	54.26	52.98	91.63	91.79	91.65	91.60	67.53	54.66	55.42	50.27
	ResNet50	88.49	89.08	88.52	88.56	61.75	66.94	61.88	60.27	91.48	91.46	91.49	91.41	76.45	68.02	67.12	63.43
	DenseNet201	83.25	83.98	83.29	83.24	39.37	45.88	39.38	35.77	90.89	91.01	90.91	90.82	64.25	51.19	50.47	45.33
	InceptionV3	53.92	53.44	53.85	53.23	17.25	17.55	17.18	15.43	90.00	89.93	90.02	89.88	8.12	1.91	3.71	1.84
	NASNetMobile	85.75	86.17	85.79	85.79	54.38	57.04	54.52	53.56	91.71	91.72	91.72	91.68	69.66	61.06	57.59	53.06
	EfficientNetB0	85.31	85.83	85.31	85.28	55.56	59.78	55.70	53.95	91.41	91.56	91.43	91.31	70.12	59.31	58.42	53.61
Proposed Loss (with SCCE)	Xception	91.39	91.41	91.31	91.36	70.71	51.05	50.22	48.82	97.74	97.63	97.91	97.76	85.11	85.46	85.21	85.18
	MobileNetV2	88.69	88.77	87.57	87.46	60.83	64.52	60.69	60.83	97.18	97.22	97.35	97.28	84.82	84.90	83.70	83.59
	ResNet50	86.53	86.44	86.53	86.46	66.01	47.14	47.18	47.16	97.13	97.01	97.34	97.17	84.90	84.81	84.90	84.83
	DenseNet201	89.15	90.40	89.41	89.80	68.41	52.53	49.64	51.04	96.63	96.79	96.70	96.74	86.13	87.38	86.39	86.78
	InceptionV3	85.30	85.14	85.31	85.22	68.34	54.68	52.54	51.78	97.20	97.45	97.16	97.30	84.37	84.80	84.32	84.29
	NASNetMobile	86.81	87.35	86.61	87.30	65.78	56.46	54.24	55.68	96.55	96.61	96.38	96.49	82.41	82.95	82.21	82.90
	EfficientNetB0	91.06	91.24	91.13	91.26	68.44	49.19	48.20	46.83	97.41	97.53	97.29	97.41	88.16	88.34	88.23	88.36
Proposed Loss (with Focal)	Xception	89.81	90.16	89.81	89.88	60.42	30.31	30.25	30.48	91.80	90.72	91.80	90.96	84.37	84.72	84.37	84.44
	MobileNetV2	88.84	89.16	88.84	88.90	58.81	27.84	27.29	28.91	87.29	87.18	87.29	87.11	84.93	84.82	83.64	83.4
	ResNet50	70.40	70.59	70.10	69.89	56.57	29.18	29.19	30.63	88.86	88.74	88.86	88.72	76.63	76.51	76.42	76.38
	DenseNet201	70.45	80.65	75.25	78.45	63.79	38.19	37.35	40.32	88.51	88.43	88.45	88.39	88.01	87.93	87.95	97.89
	InceptionV3	83.96	84.39	83.96	83.88	64.57	38.96	38.50	41.38	87.08	86.71	87.08	86.78	85.21	85.64	85.21	85.13
	NASNetMobile	81.68	76.87	78.25	75.46	62.17	36.68	35.93	38.53	87.45	86.45	87.84	86.45	83.41	82.41	83.80	82.41
	EfficientNetB0	80.52	81.46	80.52	80.63	62.28	37.06	36.51	38.87	86.81	87.32	86.81	86.79	87.56	88.07	87.56	87.54
Proposed Loss (with LDAM)	Xception	84.23	84.02	83.45	83.80	68.80	66.12	65.46	65.54	96.98	95.54	95.80	94.90	86.55	87.90	86.55	86.59
	MobileNetV2	80.40	80.60	80.23	79.24	64.42	65.34	62.54	60.68	94.93	95.18	94.46	94.38	83.27	84.71	83.27	83.11
	ResNet50	62.44	65.23	62.12	64.26	60.24	62.45	65.24	61.78	95.45	96.12	94.67	95.36	80.12	83.68	80.13	79.92
	DenseNet201	60.25	65.24	46.64	54.65	60.45	62.12	64.48	62.12	95.42	94.34	94.23	95.45	85.84	87.30	85.84	85.73
	InceptionV3	54.23	55.21	54.24	55.70	58.21	59.25	58.24	57.44	96.42	95.38	95.45	94.39	81.23	83.45	82.78	84.68
	NASNetMobile	70.41	76.56	68.90	74.56	70.67	69.56	70.24	69.78	95.69	95.45	95.44	95.45	82.24	82.34	80.45	81.65
	EfficientNetB0	70.23	85.24	78.24	81.54	54.25	55.78	58.68	56.81	95.14	94.88	95.67	93.82	89.23	89.50	89.23	88.98
Proposed Loss (with CB)	Xception	86.74	87.10	86.26	86.75	54.10	60.59	54.14	50.50	92.29	92.35	92.30	91.91	67.47	54.24	54.50	49.86
	MobileNetV2	83.79	85.10	84.59	84.32	53.95	60.04	53.05	53.00	91.65	92.27	92.12	92.20	68.11	58.32	55.66	50.78
	ResNet50	88.77	88.89	88.78	88.76	65.20	67.68	65.28	64.40	91.71	91.74	91.73	91.62	76.95	68.52	67.61	63.92
	DenseNet201	83.14	84.09	82.68	83.16	42.68	47.49	42.75	39.95	91.06	91.13	91.07	90.97	64.88	50.40	50.35	45.75
	InceptionV3	51.72	52.84	52.18	51.70	22.51	23.11	22.50	21.39	91.05	91.28	91.05	91.04	50.55	38.51	33.59	30.04
	NASNetMobile	84.98	85.48	85.03	84.96	55.82	58.29	55.80	55.00	55.82	58.28	55.79	55.00	70.44	59.81	57.20	52.60
	EfficientNetB0	85.51	86.40	85.58	85.60	56.23	60.24	56.12	55.07	91.74	91.79	91.75	91.72	70.79	61.75	58.41	54.72

Table 7: Efficiency and classification accuracy of the proposed loss (SCCE + Folk) across backbones.

Models		Xception	MobileNetV2	ResNet50	DenseNet201	InceptionV3	NASNetMobile	EfficientNetB0
No. of Parameters (Millions)		22.9	2.27	25.6	20.2	23.9	5.4	5.3
Size in MB		88	8.66	98	80	92	23	29
Time (ms) per inference step (GPU)		8.1	3.8	4.6	6.7	6.9	6.7	4.9
Acc. (%)	**CIFAR-10**	91.39	88.69	86.53	89.15	85.30	86.81	91.06
	CIFAR-100	70.71	60.83	65.79	68.41	68.34	65.78	68.44
	Fashion-MNIST	97.74	97.18	97.13	96.63	97.20	96.55	97.41
	Caltech-101	86.55	84.93	84.90	88.01	85.21	83.41	89.23

A.7 Full results across all models and datasets

This section highlights an ablation study of one model across datasets. Table 8 reports MobileNetV2 under four configurations: (i) SCCE, (ii) SCCE + the Folk penalty only, (iii) SCCE + POG/PCGrad only, and (iv) the full proposed loss (SCCE + Folk + POG/PCGrad).

Table 8: Ablation on MobileNetV2 across all four datasets.

Configuration	CIFAR-10				CIFAR-100			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
(i) SCCE	88.12	88.25	88.12	88.14	61.29	42.20	44.42	44.58
(ii) SCCE + Folk only	88.54	88.59	88.54	88.56	60.83	62.79	60.51	60.82
(iii) SCCE + POG/PCGrad only	88.36	88.42	88.21	88.29	61.34	60.27	60.44	60.34
(iv) SCCE + Folk + POG/PCGrad (proposed)	88.69	88.77	87.57	87.46	61.69	64.52	60.69	60.83
Configuration	**Fashion-MNIST**				**Caltech-101**			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
(i) SCCE	93.99	93.99	93.99	93.99	81.68	83.95	81.68	81.73
(ii) SCCE + Folk only	95.78	96.49	96.29	96.37	83.32	82.03	83.32	83.20
(iii) SCCE + POG/PCGrad only	94.22	94.94	94.51	94.72	83.49	84.93	83.49	83.40
(iv) SCCE + Folk + POG/PCGrad (proposed)	97.18	97.22	97.35	97.28	84.82	84.90	83.70	83.59

To test stability under different imbalance severities, we also run Xception on Fashion-MNIST with SCCE and SCCE + Folk across a range of imbalance factors.

Table 9: Performance across imbalance levels on Fashion-MNIST (Xception, SCCE vs. SCCE + Folk).

Loss	Imbalance factor α	Fashion-MNIST			
		Acc	Pre	Rec	F1
SCCE	0.10	94.46	94.46	94.46	94.44
	0.20	93.51	93.17	93.06	92.53
	0.40	94.06	93.69	93.63	93.15
	0.60	94.16	93.98	93.74	93.36
	0.80	95.33	92.65	92.45	92.68
SCCE + Folk	0.10	97.74	97.63	97.91	97.76
	0.20	95.42	95.98	96.45	96.42
	0.40	96.68	95.12	95.36	95.14
	0.60	96.88	95.32	95.38	95.30
	0.80	95.45	93.05	93.12	93.00

A.8 Comparison with SOTA methods

Several recent approaches tackle imbalance with modified loss functions or ensembles. Park et al. (Park et al., 2021) introduced two imbalanced CIFAR settings (exponential and step) and proposed Influence-Balanced Loss, which scales sample losses by prediction and feature norms, yielding 88.25% and 59.54% on CIFAR-10 and CIFAR-100 respectively. Chen et al. (Chen et al., 2021) used a class-balanced ensemble of ResNeXt-29 branches with a confidence-modulated loss, achieving 90.8% on CIFAR-10. Khan et al. (Khan et al., 2023) proposed Shadow Loss, a 1D triplet objective that improved generalization on CIFAR-10 (82.8%), CIFAR-100 (52.3%), and Fashion-MNIST (82.8%). GCE loss by Zhang et al. (Zhang & Sabuncu, 2018) interpolates CE and MAE, attaining 91.32% on CIFAR-10 and 94.33% on Fashion-MNIST. Other notable work includes subnetwork ensembling (Toosifar et al., 2025) (54.45% on CIFAR-100, 89.3% on Fashion-MNIST), genetic loss evolution via NGL (Akhmedova & Körber, 2024) (94.07% on Fashion-MNIST), and MGBCIL (Xian et al., 2025), a three-level incremental learner reaching 52.88% on CIFAR-100. Ren et al. (Ren et al., 2025) proposed a stepwise gradient-penalty loss that adaptively reweights cross-entropy via a power-exponential term derived from class frequencies, obtaining 71.46% on Caltech-101. Bu et al. (Bu et al., 2024) proposed an edge-enhancement preprocessing framework,

concatenating Canny edge maps with the original RGB images before training, reaching 74.39% on Caltech-101. Singhal et al. (Singhal et al., 2024) proposed a quantization-aware training framework with learnable non-uniform low-bit representations, achieving 75.27% top-1 accuracy on Caltech-101. Amelio et al. (Amelio et al., 2022) transformed trained CNNs into multilayer class-network graphs with weighted inter-layer connections, achieving 66.10% on Caltech-101.

Table 10 reports these figures alongside our own result. Since each baseline uses its own backbone, split, and training budget, this is a literature comparison, not a matched re-run under our protocol – the two should not be read as controlled against each other.

Table 10: Literature comparison. All values are accuracy (%); bold marks our own result.

CIFAR-10		**CIFAR-100**	
Method	**Acc.**	**Method**	**Acc.**
Zhang et al. (Zhang & Sabuncu, 2018)	91.32	Toosifar et al. (Toosifar et al., 2025)	54.45
Chen et al. (Chen et al., 2021)	90.81	Xian et al. (Xian et al., 2025)	52.88
Park et al. (Park et al., 2021)	88.25	Park et al. (Park et al., 2021)	59.54
Khan et al. (Khan et al., 2023)	82.82	Khan et al. (Khan et al., 2023)	52.34
SCCE + Folk + Xception	**91.39**	**SCCE + Folk + Xception**	**69.09**

Fashion-MNIST		**Caltech-101**	
Method	**Acc.**	**Method**	**Acc.**
Zhang et al. (Zhang & Sabuncu, 2018)	94.33	Ren et al. (Ren et al., 2025)	71.46
Toosifar et al. (Toosifar et al., 2025)	89.30	Bu et al. (Bu et al., 2024)	74.39
Akhmedova et al. (Akhmedova & Körber, 2024)	94.07	Singhal et al. (Singhal et al., 2024)	75.27
Khan et al. (Khan et al., 2023)	82.82	Amelio et al. (Amelio et al., 2022)	66.10
SCCE + Folk + Xception	**91.39**	**SCCE + Folk + Xception**	**85.11**

A.9 Additional visualizations

Grad-CAM. We use Grad-CAM (Selvaraju et al., 2017) on each model's final convolutional layer. Figure 5 shows that SCCE + Folk tends to localize to more accurate regions of the input than plain SCCE.



Figure 5: Grad-CAMs of each model on each dataset, using SCCE vs. SCCE + Folk.

Activation maps. Figure 6 shows sample activation maps from Xception + Focal + Folk on CIFAR-10, illustrating the hierarchical features the network relies on.

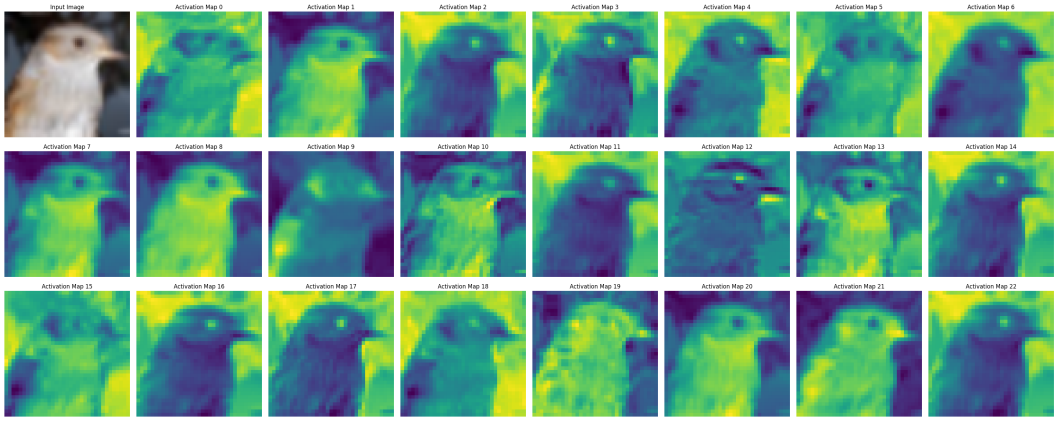

Figure 6: Activation maps of Xception + Focal + Folk on sample CIFAR-10 images.

t-SNE. Figure 7 shows t-SNE (Van der Maaten & Hinton, 2008) embeddings of test-set features on Fashion-MNIST under the proposed loss, across all seven backbones.

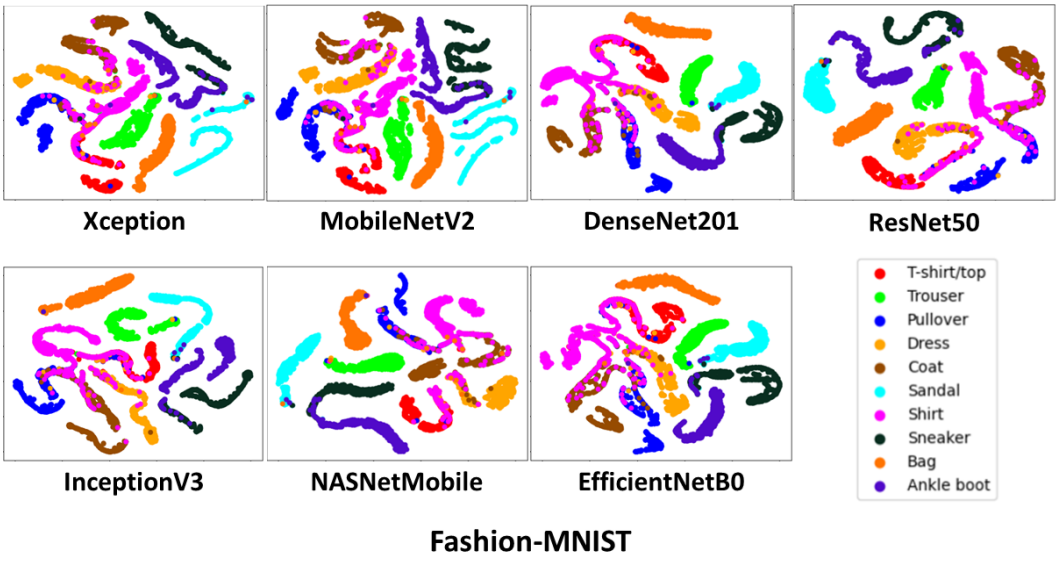

Figure 7: t-SNE plots of Fashion-MNIST test-set features using the proposed loss.