An Empirical Evaluation of Fisher-Guided Adaptive Noise for Flat Minima in Few-Shot Class-Incremental Learning

Lokesh Sathish*

Abstract

Flat minima search via noise injection during base training is a core component of F2M, a leading method for few-shot class-incremental learning (FSCIL). F2M injects isotropic Gaussian noise (constant = b for all parameters) into model parameters to encourage convergence to flat loss regions that are robust to incremental updates. I investigate whether replacing isotropic noise with Fisher Information-guided adaptive noise—scaling perturbation inversely with parameter importance—improves FSCIL performance. I term this extension A-F2M-G and evaluate it on CIFAR-100 with ResNet-18 across two optimizers and five noise configurations. My findings are negative but informative. With Adam, noise injection of any kind is redundant: all four configurations fall within 0.19 percentage points of each other. With SGD, noise injection provides a small, suggestive gain of +0.93 pp (p=0.0496), but adaptive noise offers no advantage over uniform perturbation (−0.09 pp, p=0.50). I trace this null result to a structural property of the Fisher Information distribution: a power-law skew spanning about 6 orders of magnitude (from approximately 10^{-9} at the median to 10^{-2} at the maximum) causes min-max normalization to collapse adaptive noise to effectively isotropic noise. Additionally, in my experimental setup, SGD-trained models outperformed Adam-trained models by approximately 5 percentage points in aAcc.

1 Introduction

Few-shot class-incremental learning (FSCIL) requires a model to learn new visual categories from a handful of labeled examples without forgetting previously acquired knowledge. This setting reflects realistic deployment scenarios—a deployed classifier may encounter novel objects with limited annotation budgets—and sits at the intersection of two well-studied problems: few-shot learning and continual learning. The central difficulty is catastrophic forgetting: neural networks tend to overwrite old representations when optimized on new data, and the scarcity of new-class examples exacerbates this by providing a weak learning signal that is easily dominated by forgetting.

Shi et al. [2021] proposed F2M (Flat Minima), which addresses forgetting through the geometry of the loss landscape. The key insight is that parameters residing in flat regions of the loss surface can tolerate small perturbations without significant performance degradation. F2M encourages convergence to such regions during base training by injecting isotropic Gaussian noise (constant = b for all parameters) into model parameters and averaging gradients across multiple noisy perturbations. During incremental sessions, parameters are clamped within a bound b of the learned flat minimum ϕ^*, constraining updates to remain in the flat region.

A natural question arises: if the goal is to find flat minima through noise injection, should the noise distribution be informed by parameter importance? The Fisher Information Matrix (FIM)

*Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science (BITS) Pilani, Hyderabad Campus. Correspondence: brodie1of1@gmail.com

provides a principled measure of how sensitive a model's predictions are to changes in each parameter. Elastic Weight Consolidation [Kirkpatrick et al., 2017] uses diagonal Fisher values to penalize updates to important parameters during continual learning. I hypothesized that applying a similar principle to noise injection—perturbing important parameters less and unimportant parameters more—would produce flatter, more robust minima than uniform perturbation.

I term this extension A-F2M-G (Adaptive Flat Minima with Gaussian prototypes) and evaluate it with the noise scaling formula $\sigma = b(1 - \alpha\,\tilde{\omega})$, where $\tilde{\omega}$ is the min-max normalized diagonal Fisher importance per layer and α controls the dynamic range. When $\alpha = 0.9$, the noise standard deviation spans a 10:1 range across parameters; when $\alpha = 0.5$, a 2:1 range.

My results do not support the hypothesis. Under Adam optimization, noise injection of any kind—uniform, adaptive, or none—produces statistically indistinguishable FSCIL accuracy. Under SGD, noise injection gives a small, suggestive gain (+0.93 pp, $p < 0.05$), but adaptive noise performs no better than uniform perturbation. I investigate why through an analysis of the Fisher Information distribution and identify a structural explanation: the diagonal Fisher values follow a power-law distribution with a max-to-median ratio exceeding 2.6 million. After min-max normalization, the vast majority of parameters receive $\tilde{\omega}$ close to zero, making adaptive noise functionally identical to isotropic noise.

My contributions are as follows. First, I provide controlled experiments showing that Fisher-guided adaptive noise does not improve over uniform noise for flat minima search in FSCIL. Second, I demonstrate that the choice of base training optimizer (Adam vs. SGD) has a larger effect on FSCIL performance than the noise injection strategy, with SGD outperforming Adam by approximately 5 percentage points in my setup. Third, I provide a structural explanation for the adaptive noise null result grounded in the empirical distribution of Fisher Information values.

2 Related Work

Few-Shot Class-Incremental Learning. FSCIL methods broadly fall into three categories: representation-based, structure-based, and optimization-based approaches. Tao et al. [2020] proposed TOPIC, which uses a neural gas network to preserve the topology of feature space during incremental updates. Zhang et al. [2021] introduced CEC, which decouples feature extraction from classification by learning a graph-based classifier over frozen features. FACT [Zhou et al., 2022] learns a set of virtual prototypes during base training that reserve feature space for future classes. ALICE [Peng et al., 2022] uses angular penalty losses to learn more discriminative base features. SAVC [Song et al., 2023] extends the virtual prototype idea with semantic-aware generation. F2M [Shi et al., 2021] takes an optimization-based approach, seeking flat minima during base training so that constrained incremental updates do not degrade performance. My work builds directly on F2M by investigating whether its noise injection mechanism can be improved with Fisher Information guidance.

Flat Minima and Sharpness-Aware Optimization. The connection between flat minima and generalization has a long history [Keskar et al., 2017]. Sharpness-Aware Minimization (SAM) [Foret et al., 2021] explicitly seeks parameters that minimize both the loss value and its sharpness by performing a gradient ascent step before each gradient descent step. F2M's noise injection approach is related but distinct: rather than adversarially perturbing parameters, it samples random perturbations and averages the resulting gradients. This can be seen as a stochastic approximation to the expected loss over a neighborhood of the current parameters. My investigation of adaptive noise extends this by asking whether the neighborhood should be anisotropic, shaped by parameter importance.

Fisher Information in Continual Learning. The diagonal Fisher Information Matrix has been widely used as a parameter importance measure in continual learning. Kirkpatrick et al. [2017] introduced Elastic Weight Consolidation (EWC), which adds a quadratic penalty weighted by diagonal Fisher values to prevent forgetting. Subsequent work has refined this idea through online Fisher accumulation and alternative importance measures. My use of Fisher Information differs from EWC: rather than penalizing parameter updates, I use Fisher values to modulate the noise injection during base training. The goal is to perturb unimportant parameters more aggressively while protecting important ones, potentially producing flatter and more informative minima.

3 Method

3.1 F2M Background

F2M [Shi et al., 2021] trains a feature extractor f_θ with a linear classifier on base classes, then performs incremental learning using a nearest-mean classifier (NMC) with prototype representations. During base training, F2M injects isotropic Gaussian noise (constant $= \mathrm{b}$ for all parameters) into model parameters and computes gradients at multiple noisy perturbations per minibatch, averaging these gradients before applying the optimizer step. This encourages the optimizer to find parameters that perform well not just at a single point but across a neighborhood, which corresponds to a flat region of the loss surface.

After base training, the learned parameters ϕ^* define a flat minimum. During each incremental session, the model receives a 5-way 5-shot support set of novel classes. The feature extractor is fine-tuned with a prototype-based loss while parameters are clamped within a bound b of ϕ^*, ensuring the model remains in the flat region. Prototypes for new classes are computed as class-mean feature vectors using the support set, and classification is performed via cosine similarity between test features and all accumulated prototypes.

3.2 A-F2M-G: Fisher-Scaled Adaptive Noise

I extend F2M by replacing isotropic noise with Fisher Information-guided adaptive noise. The diagonal Fisher Information ω_i for parameter θ_i measures how sensitive the model's output distribution is to perturbations of θ_i:

$$\omega_i = \mathbb{E}_{x\sim\mathcal{D}}\left[\left(\frac{\partial \log p(y|x;\theta)}{\partial \theta_i}\right)^2\right]. \tag{1}$$

I compute ω once after the warmup phase using 5000 randomly sampled training images with the model in evaluation mode. The Fisher values are then used to scale the noise injection during the A-FMS (Adaptive Flat Minima Search) phase.

For each layer ℓ of the feature extractor, I apply min-max normalization:

$$\tilde{\omega}_i^{(\ell)} = \frac{\omega_i^{(\ell)} - \min_j \omega_j^{(\ell)}}{\max_j \omega_j^{(\ell)} - \min_j \omega_j^{(\ell)} + \epsilon}, \tag{2}$$

where $\epsilon = 10^{-8}$ prevents division by zero. The noise standard deviation for each parameter is then:

$$\sigma_i = b \cdot (1 - \alpha \cdot \tilde{\omega}_i), \tag{3}$$

where b is the flat region bound and $\alpha \in [0, 1]$ controls the dynamic range. When $\alpha = 0.9$, important parameters ($\tilde{\omega} \approx 1$) receive noise with $\sigma = 0.1b$ while unimportant parameters ($\tilde{\omega} \approx 0$) receive $\sigma = b$,

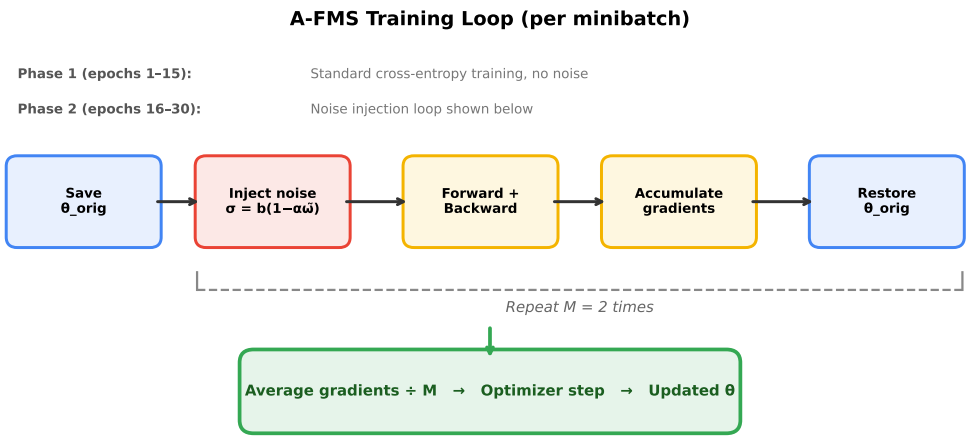

Figure 1: Overview of the A-FMS training loop during Phase 2 (epochs 16–30). For each minibatch, original parameters are saved, noise is injected and gradients accumulated across M=2 perturbations, parameters are restored, and a single optimizer step is applied with averaged gradients.

a 10:1 ratio. The classifier head, for which no Fisher Information is computed, receives isotropic noise at scale b.

I also evaluate a log-scale normalization variant:

$$\tilde{\omega}_i^{(\ell)} = \frac{\log(\omega_i^{(\ell)} + \epsilon) - \min_j \log(\omega_j^{(\ell)} + \epsilon)}{\max_j \log(\omega_j^{(\ell)} + \epsilon) - \min_j \log(\omega_j^{(\ell)} + \epsilon) + \epsilon}, \tag{4}$$

which spreads the Fisher distribution more evenly before normalization, giving more parameters a meaningfully reduced noise level.

3.3 Training Protocol

Base training proceeds in two phases over 30 epochs. During the warmup phase (epochs 1–15), the model is trained with standard cross-entropy loss without noise injection. At the boundary between epochs 15 and 16, the diagonal Fisher Information is computed once using 5000 samples from the base training set. During the A-FMS phase (epochs 16–30), each minibatch is processed as follows: the original parameters are saved, noise is injected $M = 2$ times, gradients are accumulated across both perturbations from all trainable parameters in the feature extractor and classifier head, original parameters are restored, the accumulated gradients are averaged by M, and a single optimizer step is applied.

For incremental sessions, I follow the standard FSCIL protocol. The feature extractor is fine-tuned for 10 epochs with SGD (learning rate 0.01, momentum 0.9) using a prototype-based metric loss. Parameters are clamped within b of ϕ^* after each optimizer step. BatchNorm running statistics are frozen during fine-tuning and restored from ϕ^* before prototype computation. Class prototypes are computed as L2-normalized mean feature vectors, with a re-normalization step after averaging to ensure unit norm. Evaluation uses cosine similarity between L2-normalized test features and prototypes across all seen classes.

Table 1: Adam optimizer study on CIFAR-100 ($b = 0.005$, 30 epochs). aAcc is the mean accuracy across all 9 sessions. S0 and S8 denote session 0 (base) and session 8 (final) accuracy. PD is performance drop (S0 − S8). Results are mean ± std over 3 seeds.

Exp	Noise Type	aAcc (%)	S0 (%)	S8 (%)	PD (pp)
E1	None	63.25 ± 0.57	78.03	52.28	25.75
E2	Uniform	63.44 ± 0.13	78.34	52.18	26.17
E3	Adaptive (α=0.9)	63.29 ± 0.34	78.11	52.04	26.07
E4	Adaptive (α=0.5)	63.39 ± 0.38	78.06	52.36	25.70
E3 vs. E1: $\Delta = +0.04$ pp, $p = 0.90$ (n.s.)			E3 vs. E2: $\Delta = -0.15$ pp, $p = 0.39$ (n.s.)		

4 Experiments

4.1 Setup

I evaluate on CIFAR-100 [Krizhevsky, 2009] using a ResNet-18 [He et al., 2016] backbone pretrained on ImageNet. Following the standard FSCIL protocol, I use 60 base classes for training and 40 novel classes distributed across 8 incremental sessions of 5-way 5-shot. Images are resized to 84×84. All experiments are run across three seeds $\{42, 123, 456\}$ with deterministic CUDA settings, and I report mean ± standard deviation. Statistical comparisons use paired t-tests across seeds.

I evaluate five noise configurations spanning two base training optimizers. For the Adam study, I use Adam (learning rate 10^{-3}, weight decay 10^{-4}) with bound $b = 0.005$ and cosine annealing. For the SGD study, I use SGD (learning rate 10^{-2}, momentum 0.9) with bound $b = 0.01$, matching the original F2M optimizer choice more closely. All runs use $M = 2$ noise samples per minibatch during A-FMS.

4.2 Study 1: Adam Optimizer

Table 1 presents results under Adam optimization. The four configurations—no noise, uniform noise, adaptive noise (α=0.9), and adaptive noise (α=0.5)—produce average accuracies spanning only 0.19 percentage points. Experiment E3 (adaptive, α=0.9) achieves 63.29% aAcc compared to 63.25% for E1 (no noise), a difference of +0.04 pp with $p = 0.90$. The comparison between adaptive and uniform noise (E3 vs. E2) yields −0.15 pp ($p = 0.39$). Neither comparison approaches statistical significance.

The session 0 accuracies cluster near 78% and the session 8 accuracies near 52%, with performance drops of approximately 26 pp across all configurations. The low variance of E2 (±0.13) and higher variance of E1 (±0.57) may suggest that noise injection slightly stabilizes training, but the effect is negligible in magnitude and $n = 3$ is insufficient to draw conclusions about variance. The evidence suggests that under Adam optimization, noise injection during base training is functionally redundant.

4.3 Study 2: SGD Optimizer

Table 2 presents results under SGD optimization. Here the picture changes. E2-equivalent S2 (uniform noise) achieves 69.12% aAcc, a gain of +0.93 pp over S1 (no noise, 68.19%) with $p = 0.0496$. With only $n = 3$ seeds, this p-value should be considered suggestive rather than definitive. Adaptive noise with min-max normalization (S3, α=0.9) achieves 69.03%, a gain of +0.84 pp over S1 ($p = 0.018$).

Table 2: SGD optimizer study on CIFAR-100 ($b = 0.01$, lr $= 0.01$, momentum $= 0.9$). Significance markers: $^*p < 0.05$ (paired t-test vs. S1). All values are mean ± std over 3 seeds.

Exp	Noise Type	α	aAcc (%)	S0 (%)	S8 (%)	PD (pp)
S1	None	—	68.19 ± 0.29	81.46	57.95	23.51
S2	Uniform	—	$\mathbf{69.12 \pm 0.14}^*$	**82.51**	58.76	23.75
S3	Adaptive (minmax)	0.9	$69.03 \pm 0.10^*$	82.42	58.77	23.65
S4	Adaptive (minmax)	0.5	69.09 ± 0.07	82.37	58.85	23.52
S5	Adaptive (log)	0.9	68.83 ± 0.02	82.17	58.58	23.58

S2 vs. S1: $\Delta = +0.93$ pp, $p = 0.0496^*$ S3 vs. S1: $\Delta = +0.84$ pp, $p = 0.018^*$

S3 vs. S2: $\Delta = -0.09$ pp, $p = 0.50$ (n.s.) S5 vs. S3: $\Delta = -0.20$ pp, $p = 0.04^*$ (log worse than minmax)

S4 vs. S1 not tested (intermediate α variant).

However, the comparison that matters most—S3 vs. S2 (adaptive vs. uniform)—shows only -0.09 pp ($p = 0.50$). Adaptive noise provides no advantage over uniform perturbation.

The log-scale normalization variant (S5) performs slightly worse than both uniform and min-max adaptive noise: -0.30 pp relative to S2 ($p = 0.07$) and -0.20 pp relative to S3 ($p = 0.04$). Log-scale normalization spreads the Fisher distribution more evenly, giving more parameters a meaningfully reduced noise level, but this additional protection of important parameters appears to slightly hurt rather than help. The standard deviations under SGD are notably small (S4: ± 0.07, S5: ± 0.02); the lower variance of noise-injection runs (e.g., S5: ± 0.02) may reflect stabilization, though $n = 3$ is insufficient to draw conclusions about variance.

4.4 Optimizer Comparison

A striking finding across both studies is the large gap between optimizers. SGD consistently outperforms Adam by approximately 5 percentage points in aAcc: the no-noise SGD baseline (S1, 68.19%) already exceeds the best Adam configuration (E2, 63.44%) by 4.75 pp. With noise, the gap widens: S2 (69.12%) vs. E2 (63.44%) is a 5.68 pp difference. Session 0 accuracy is similarly higher under SGD (81–82% vs. 78%), as is session 8 accuracy (58–59% vs. 52%), indicating that SGD produces better base representations that also retain better under incremental learning. Performance drop is slightly lower under SGD (23–24 pp vs. 25–26 pp), suggesting modestly better stability as well.

This comparison should be interpreted with caution, as the two optimizer studies used different learning rates (10^{-3} vs. 10^{-2}), different bounds ($b = 0.005$ vs. 0.01), and the gap may partially reflect these hyperparameter differences rather than a fundamental optimizer effect.

5 Analysis

5.1 Why Noise is Irrelevant with Adam

I hypothesize that Adam's per-parameter adaptive learning rates, which scale updates based on gradient history, may provide an implicit form of loss landscape smoothing that makes explicit noise injection redundant. Testing this hypothesis directly would require ablation studies on Adam's internal mechanisms, which is beyond the scope of this work. When the optimizer already modulates its step sizes based on gradient history, external noise injection provides no additional signal about

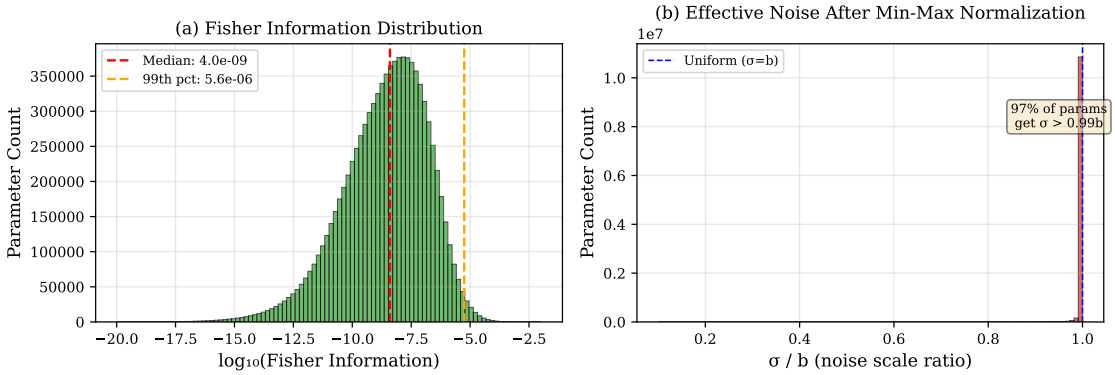

Figure 2: Distribution of diagonal Fisher Information values across 9.1M feature extractor parameters (left) and resulting effective noise standard deviation after min-max normalization with $\alpha = 0.9$ (right). The extreme power-law skew causes min-max normalization to assign $\tilde{\omega} \approx 0$ to the vast majority of parameters, making adaptive noise functionally identical to isotropic noise.

the loss landscape geometry. This may explain why all four Adam configurations in Table 1 produce virtually identical results: Adam's internal mechanics may subsume the role of noise injection.

This observation has practical implications. Researchers using Adam or similar adaptive optimizers (AdamW, AdaGrad) for FSCIL base training can likely omit the noise injection phase entirely, saving computational overhead without sacrificing performance.

5.2 Why Noise Type Does Not Matter with SGD

The null result between adaptive and uniform noise under SGD requires a different explanation, since SGD does benefit from noise injection. I identify two contributing factors.

Fisher Distribution Skew. Figure 2 shows the distribution of diagonal Fisher values across the 9.1 million feature extractor parameters. The distribution follows a power law spanning about 6 orders of magnitude (from approximately 10^{-9} at the median to 10^{-2} at the maximum). The max-to-median ratio exceeds 2.6 million. Key percentiles are: 50th at 3.96×10^{-9}, 95th at 9.05×10^{-7}, 99th at 5.64×10^{-6}, and 99.9th at 4.21×10^{-5}.

After min-max normalization within each layer, the vast majority of parameters receive $\tilde{\omega}$ close to zero. Substituting into Equation 3 with $\alpha = 0.9$: for most parameters, σ is functionally indistinguishable from the isotropic noise level $\sigma = b$. Only the extreme tail of the Fisher distribution (the top 0.1–1% of parameters per layer) receives meaningfully reduced noise. The adaptive mechanism is structurally incapable of producing behavior that differs from isotropic noise for the vast majority of parameters.

Gradient Averaging as Low-Pass Filter. The $M = 2$ gradient averaging step in A-FMS introduces additional smoothing. Each minibatch computes gradients at two independently sampled noisy parameter configurations and averages them. I hypothesize that this averaging may act as a low-pass filter on the noise-specific gradient signal. Even if adaptive noise produced a meaningfully different perturbation pattern from isotropic noise at the individual sample level, the averaging step would attenuate the difference. The flat minima search mechanism finds its solution through the averaging process itself—the stochastic exploration of the loss neighborhood—rather than through the specific shape of the perturbation distribution.

5.3 Log-Scale Normalization

The log-scale variant (S5) was designed to address the Fisher skew problem by compressing the dynamic range before normalization. In log space, the Fisher distribution is more evenly spread, and a larger fraction of parameters receives a meaningfully reduced $\tilde{\omega}$. However, this did not help. S5 (68.83%) performs slightly worse than S3 (69.03%), with the difference reaching significance ($p = 0.04$). More aggressive protection of high-Fisher parameters appears to reduce the effectiveness of the flat minima search, possibly by over-constraining the noise in directions that are important for finding good flat regions. The result reinforces the conclusion that the specific noise distribution matters less than the mere presence of stochastic perturbation.

5.4 Differences from Original F2M

Several differences between my setup and the original F2M implementation of Shi et al. [2021] should be noted. First, F2M injects noise only into the last 4–8 convolutional layers, while I inject into all trainable parameters including the classifier head. Second, F2M includes a "prototype fixing" loss term during base training that I do not implement. Third, my incremental sessions use 10 epochs at learning rate 0.01 compared to F2M's 6 epochs at 0.02. Fourth, F2M does not fully specify the base training hyperparameters (epochs, learning rate, schedule) in its main paper. These differences mean my absolute accuracy numbers are not directly comparable to those reported by Shi et al. [2021], but my controlled comparisons between noise types remain valid since all configurations share the same experimental protocol.

6 Discussion and Limitations

6.1 Comparison to Published Methods

Table 3 places my results in context with published FSCIL methods on CIFAR-100. My best SGD configuration (S2, uniform noise) achieves 82.51% session 0 accuracy and 69.12% aAcc, which exceeds all listed baselines in raw numbers, though the different backbones, image resolutions, and training protocols preclude direct comparison. However, direct comparison should be interpreted with caution due to the implementation differences described in Section 5.4. My session 0 accuracy is notably higher than most published methods, which may reflect differences in base training duration, data augmentation, or the use of full-model noise injection rather than layer-selective injection. A rigorous comparison would require replicating each baseline under identical conditions, which is beyond the scope of this work.

6.2 Limitations

This study has several limitations that constrain the generality of my conclusions. I evaluate on a single dataset (CIFAR-100) with one backbone (ResNet-18). FSCIL benchmarks also include miniImageNet and CUB-200, and my findings may not transfer to these settings, particularly CUB-200 where fine-grained visual distinctions could interact differently with noise injection. I use 3 random seeds per configuration, while Shi et al. [2021] report results over 10 runs, giving them tighter confidence intervals.

My noise injection covers all trainable parameters, whereas F2M targets only the last 4–8 convolutional layers. It is possible that Fisher-guided noise would show a benefit when applied selectively to later layers where the Fisher distribution may be less skewed. I also omit F2M's

Table 3: Comparison with published FSCIL methods on CIFAR-100. My setup uses 84×84 images with ImageNet-pretrained weights and a different training protocol from the baselines; the baselines use varying backbones (often ResNet-20 trained from scratch) and training protocols, so these numbers are not directly comparable and are included only for rough context. Published numbers are from the respective original papers.

Method	Venue	S0 (%)	S8 (%)	aAcc (%)
F2M	NeurIPS 2021	71.33	47.55	∼58
CEC	CVPR 2021	73.07	49.14	∼60
FACT	CVPR 2022	74.60	52.10	∼62
ALICE	ECCV 2022	79.00	54.10	∼64
SAVC	CVPR 2023	77.90	54.20	∼65
Ours (S2, SGD+uniform)	—	**82.51**	**58.76**	**69.12**

prototype fixing loss during base training, which could interact with the noise mechanism in ways I do not capture.

The Fisher Information is computed once at epoch 15 from 5000 samples. A different computation strategy—online updates, more samples, or computation at a different training stage—might yield a less skewed distribution, though the fundamental power-law nature of Fisher values in deep networks is well-documented.

Finally, I evaluate only $M = 2$ noise samples per minibatch. Larger M values might amplify differences between noise distributions by reducing the low-pass filtering effect, though at increased computational cost.

7 Conclusion

I investigated whether Fisher Information-guided adaptive noise improves flat minima search for FSCIL over the uniform noise injection used in F2M. The answer is no. Under Adam optimization, noise injection of any kind is redundant. Under SGD, noise injection provides a suggestive improvement of approximately 0.9 percentage points ($p = 0.0496$, $n = 3$), but the type of noise—uniform, adaptive with min-max Fisher normalization, or adaptive with log-scale Fisher normalization—does not matter. Uniform noise is the simplest and performs as well as or better than any adaptive variant.

A likely structural reason is the extreme skew of the diagonal Fisher Information distribution. With values spanning about 6 orders of magnitude (from approximately 10^{-9} at the median to 10^{-2} at the maximum) and a max-to-median ratio exceeding 2.6 million, min-max normalization collapses the adaptive noise formula to effectively isotropic perturbation for the vast majority of parameters. Log-scale normalization spreads the distribution but slightly hurts performance by over-protecting important parameters.

An ancillary finding is that in my setup, SGD outperformed Adam by approximately 5 percentage points despite Adam's theoretical connection to adaptive geometry. This optimizer effect dwarfs any effect of the noise injection strategy and warrants further investigation.

For practitioners working on FSCIL, my results suggest that the noise injection mechanism in F2M is worth retaining when using SGD but does not benefit from Fisher Information guidance. Computational resources are better spent on other aspects of the training pipeline.

AI Disclosure

AI tools (Kiro) were used to assist with code implementation, experiment automation, data analysis, and initial manuscript drafting. All experimental results were generated from original code executed on my hardware. I reviewed, verified, and substantially revised all AI-assisted content, including checking every cited reference against its original source. Responsibility for the final manuscript, including any errors, rests entirely with me.

References

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C. Lovell. Few-shot class-incremental learning from an open-set perspective. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 382–397, 2022.

Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 6747–6761, 2021.

Zeyin Song, Yifan Zhao, Yujun Shi, Peixi Peng, Li Yuan, and Yonghong Tian. Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24183–24192, 2023.

Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12183–12192, 2020.

Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12455–12464, 2021.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9046–9056, 2022.