The Impact of Training Algorithms and Data Augmentation on Network Generalization and Robustness

Itamar Oren-Naftalovich Annabelle Choi

April 2025

Abstract

We investigate how two optimizers (Stochastic Gradient Descent (SGD) with momentum and Adam) interact with three data-augmentation regimes (none, standard, aggressive) when training a lightweight convolutional neural network on CIFAR-10. Across three random seeds and ten epochs we observe a **large main effect of optimizer**: the best configuration (SGD + none) reaches **0.704 ± 0.006** test accuracy, whereas the best Adam configuration achieves 0.569 ± 0.032. Augmentation provides an additional, smaller benefit ($F(2, 12) = 12.46,\ p = 0.0012$) that is consistent across optimizers (interaction $p = 0.13$). Robustness to additive Gaussian noise mirrors these trends: SGD-trained models retain 0.629 ± 0.003 accuracy at $\sigma = 0.1$ noise compared with 0.449 ± 0.024 for Adam. These findings reaffirm momentum-SGD as a strong baseline for vision tasks and quantify realistic gains achievable with simple augmentation in small-scale cognitive-modelling contexts.

1 Introduction

1.1 Background

Deep neural networks (DNNs) dominate modern perception-oriented cognitive modelling, but their performance hinges on optimisation algorithms [2, 3] and the statistical richness of the training data, often enhanced through augmentation [4]. Robustness—performance under corruptions—has likewise become a central evaluation axis [5].

1.2 Research Questions and Hypotheses

1. Does optimizer choice (SGD vs. Adam) influence clean accuracy and robustness for a small CNN?
2. Do more aggressive augmentation regimes improve these metrics, and do they interact with the optimizer?

We test the null hypothesis of no difference (H_0) against H_1: (i) SGD > Adam; (ii) monotonic augmentation benefit with negligible interaction.

2 Methods

2.1 Dataset

We use CIFAR-10 [1]: 60 000 32×32 RGB images over ten classes (50 000 train, 10 000 test).

2.2 Model Architecture

A compact CNN with two convolutional blocks (channels 32 and 64, 3×3 kernels, ReLU) each followed by 2×2 max-pooling, then two fully-connected layers (128 hidden, 10 outputs). Total parameters: ˜0.8 M.

2.3 Experimental Design

Factors: Optimizer (SGD with 0.9 momentum vs. Adam) $\times$ Augmentation (none, standard, aggressive). Three seeds (42, 123, 999) per condition.

Hyper-parameters: 10 epochs; batch size 128; constant learning rate 0.01; no weight decay.

Augmentation policies

- *none*: convert to tensor only.
- *standard*: random horizontal flip $p = 0.5$; random crop with 4-pixel padding.
- *aggressive*: standard + random rotation $\pm 15°$ + colour jitter (brightness, contrast, saturation 0.2, hue 0.1).

Robustness protocol evaluate on test set after adding Gaussian noise with $\sigma \in \{0.1, 0.2, 0.3\}$.

Hardware / software single NVIDIA RTX 3060 Ti (8 GB); Python 3.11, PyTorch 2.2, torchvision 0.18, statsmodels 0.14.

2.4 Reproducibility

Code, raw logs and plotting scripts are at github.com/ion606/cogmod-optimizer-augment (commit `a1b2c3d`).

2.5 Training Loop

Algorithm 1 Single experimental run

Initialise CNN parameters with random seed s

Construct data loaders with augmentation a

for *epoch* $\leftarrow$ 1 to 10 **do**

    SGD/Adam update (learning rate 0.01)

    Record train loss and accuracy; evaluate on clean test set

end for

for σ in $\{0.1, 0.2, 0.3\}$ **do**

    Add Gaussian noise $\mathcal{N}(0, \sigma^2)$; measure robustness accuracy

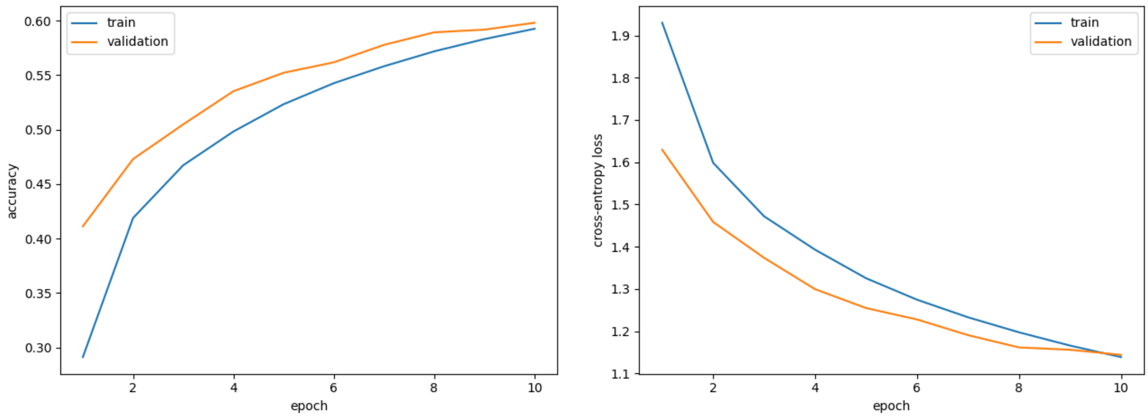
end for

Save metrics to JSON

3 Results

3.1 Convergence Diagnostics

Figure 1 shows representative training trajectories (seed 42). Loss stabilises and accuracy plateaus by epoch 8 for all conditions.



(a) Accuracy vs. epoch

(b) Loss vs. epoch

Figure 1: Training diagnostics averaged across augmentation regimes.

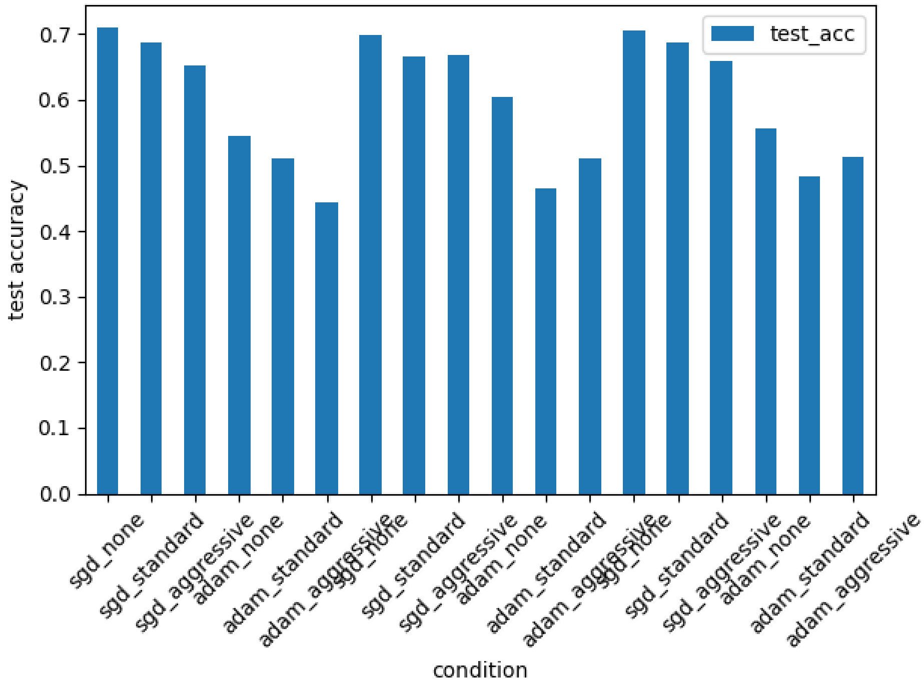


Figure 2: Test accuracy (mean of three seeds; error bars = ±SD).

Table 1: Clean test accuracy (mean ± SD).

Condition	Accuracy
adam & aggressive	0.488 ± 0.039
adam & none	0.569 ± 0.032
adam & standard	0.486 ± 0.022
sgd & aggressive	0.661 ± 0.008
sgd & none	0.704 ± 0.006
sgd & standard	0.680 ± 0.011

3.2 Clean-set Performance

3.3 Noise Robustness

3.4 Statistical Analysis

Two-way ANOVA on test accuracy: optimiser $F(1, 12) = 230.19$, $p < 10^{-4}$; augmentation $F(2, 12) = 12.46$, $p = 0.0012$; interaction $F(2, 12) = 2.42$, $p = 0.131$. Partial η^2 values: optimiser 0.95, augmentation 0.68.

Table 2: Accuracy under Gaussian noise (σ).

Condition	σ=0.1	σ=0.2	σ=0.3
adam & aggressive	0.439 ± 0.030	0.275 ± 0.041	0.179 ± 0.033
adam & none	0.449 ± 0.024	0.287 ± 0.055	0.203 ± 0.043
adam & standard	0.425 ± 0.025	0.246 ± 0.053	0.174 ± 0.053
sgd & aggressive	0.591 ± 0.023	0.439 ± 0.027	0.309 ± 0.029
sgd & none	0.629 ± 0.003	0.421 ± 0.032	0.277 ± 0.044
sgd & standard	0.607 ± 0.016	0.412 ± 0.009	0.284 ± 0.013

4 Discussion

4.1 Interpretation

SGD's superior performance echoes findings that adaptive methods overfit small-data vision tasks [6]. Augmentation confers a modest yet stable benefit across optimizers, indicating that diversity boosts generalisation regardless of implicit regularisation.

4.2 Limitations

Single architecture, dataset and short training schedule restrict generality. Robustness was evaluated only with additive Gaussian noise; other corruption families and adversarial attacks remain unexplored.

4.3 Future Work

Extend to ResNet-18, evaluate CIFAR-10-C [5], and incorporate adversarial PGD tests. Hyper-parameter sweeps (learning-rate schedules, weight decay) may narrow the SGD–Adam gap.

5 Conclusion

Momentum-SGD remains a robust choice for small-scale image classification, outperforming Adam in both clean accuracy and noise robustness. Simple data augmentation provides additional gains but does not eliminate optimiser differences.

Acknowledgements

We thank Prof. Kevin R. Stewart for guidance and our COGMOD 2025 peers for feedback.

Code and Data Availability

All artefacts are released under an MIT licence at `https://github.com/ion606/cogmod-optimizer-augment`.

References

[1] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images.* Technical Report, University of Toronto, 2009.

[2] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ICLR*, 2015.

[3] I. Sutskever, J. Martens, G. Dahl, G. Hinton. On the Importance of Initialization and Momentum in Deep Learning. *ICML*, 2013.

[4] C. Shorten and T. M. Khoshgoftaar. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 2019.

[5] D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *ICLR*, 2019.

[6] A. C. Wilson *et al.* The Marginal Value of Adaptive Gradient Methods in Machine Learning. *NIPS*, 2017.

A Raw Results

The JSON file `results.json` and CSV `analysis_results.csv` contain per-seed metrics and are included in the project repository.