Vol.2 No.2 October 2024: 49-58 e-ISSN: 3064-4372
International Journal of Computer Science and Humanitarian AI (IJCSHAI) DOI: 10.21512/ijcshaijournal.v2i2.14533

Adaptive Gradient Compression: An Information-Theoretic Analysis of Entropy and Fisher-Based Learning Dynamics

Hidayaturrahman

Computer Science Department, School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
hidayaturrahman@binus.ac.id

***Abstract*—Deep neural networks require intensive computation and communication due to the large volume of gradient updates exchanged during training. This paper investigates Adaptive Gradient Compression (AGC), an information-theoretic framework that reduces redundant gradients while preserving learning stability. Two independent compression mechanisms are analyzed: an entropy-based scheme, which filters gradients with low informational uncertainty, and a Fisher-based scheme, which prunes gradients with low sensitivity to the loss curvature. Both approaches are evaluated on the CIFAR-10 dataset using a ResNet-18 model under identical hyperparameter settings. Results show that entropy-guided compression achieves a 33.8× reduction in gradient density with only a 4.4% decrease in test accuracy, while Fisher-based compression attains 14.3× reduction and smoother convergence behavior. Despite modest increases in per-iteration latency, both methods maintain stable training and demonstrate that gradient redundancy can be systematically controlled through information metrics. These findings highlight a new pathway toward information-aware optimization, where learning efficiency is governed by the informational relevance of gradients rather than their magnitude alone. Furthermore, this study emphasizes the practical significance of integrating information theory into deep learning optimization. By selectively transmitting gradients that carry higher information content, AGC effectively mitigates communication bottlenecks in distributed training environments. Experimental analyses further reveal that adaptive compression dynamically adjusts to training dynamics, providing robustness across various learning stages. The proposed framework can thus serve as a foundation for developing future low-overhead optimization methods that balance accuracy, stability, and efficiency, and crucial aspects for large-scale deep learning deployments in edge and cloud computing contexts.**

Received: Oct. 10, 2025; received in revised form: Oct. 16,2025; accepted: Nov. 01,2025; available online: Nov. 01,2025.

Corresponding: hidayaturrahman@binus.ac.id

Keywords—Gradient compression; entropy; Fisher information; learning dynamics; information theory; optimization efficiency; deep learning

I. INTRODUCTION

The rapid advancement of deep neural networks (DNNs) has brought unprecedented progress in computer vision, natural language processing, and other domains of artificial intelligence. However, this success comes at the cost of high computational and energy demands, primarily due to the enormous volume of gradient updates propagated through millions or even billions of parameters during training. [1] In many cases, a substantial portion of these gradients contributes negligibly to the overall improvement of model performance, resulting in significant redundancy within the learning dynamics. [2]

Gradient compression techniques have emerged as a promising direction to mitigate this inefficiency. Prior research has primarily focused on reducing communication overhead in distributed or federated learning environments by compressing or quantizing gradients before transmission. [3] While these methods achieve notable efficiency gains, they tend to view gradients as mere numerical quantities, without analyzing their underlying information content. Consequently, little is known about how different forms of compression affect the information flow and representation dynamics of neural networks during training.

This study explores the concept of Adaptive Gradient Compression (AGC) as an analytical framework to understand and control redundancy in gradient-based learning. Instead of relying solely on magnitude-based thresholding, AGC incorporates information-theoretic measures—specifically, entropy and Fisher information—to guide compression. The intuition is that entropy captures the uncertainty or dispersion of gradient distributions, while Fisher information quantifies the sensitivity of the model parameters to perturbations in the loss landscape. [4], [5]

Copyright © 2025

The primary objective of this research is to analyze the learning dynamics induced by entropy-based and Fisher-based compression mechanisms, both applied independently. Through controlled experiments on CIFAR-10, we measure not only traditional performance metrics such as accuracy and training loss, but also the structural properties of the gradients themselves—namely, sparsity, non-zero ratio (NNZ), and step time. These analyses provide insight into how information-guided compression modulates the efficiency and stability of the training process.

The contributions of this work are threefold. First, it presents a systematic analysis of entropy- and Fisher-based compression methods within the same adaptive framework, providing a unified view of how different information measures influence optimization.

Second, it offers quantitative and visual evidence that supports the hypothesis that gradient redundancy is not random but structurally correlated with information flow across training epochs. [6]

Communication overhead has been a primary bottleneck in data-parallel training, motivating compressive techniques that reduce gradient payload without destabilizing convergence. Early quantization methods, such as QSGD [7], provide unbiased stochastic quantization with theoretical guarantees, while Deep Gradient Compression (DGC) combines sparsification, momentum correction, and local gradient clipping to deliver large bandwidth reductions in practice [8]. Prior work on 1-bit SGD demonstrated that aggressive quantization can scale speech DNN training across commodity GPUs with minimal accuracy loss [9]. Subsequent studies on structured sparsification and sketching further formalized convergence under compressed updates [10], establishing a foundation for modern gradient compression pipelines.

A central theme in sparsification is preserving optimizer dynamics despite dropping most coordinates. Top-k and threshold-based sparsification reduce communicated entries but risk bias accumulation; residual accumulation / memory mechanisms (e.g., Sparsified SGD with Memory) re-inject dropped mass in future steps to recover convergence rates [11]. Empirically, Sparse Communication for Distributed SGD showed substantial traffic reduction without compromising BLEU in NMT [5]. Theoretically, Error Feedback Fixes SignSGD proved that adding an error-feedback buffer restores descent directions even for biased compressors, stabilizing a range of schemes (sign, top-k, quant) [12].

Sign-based updates achieve extreme compression by communicating only the sign of coordinates, sometimes with majority vote aggregation to mitigate noise [13]. Unbiased or variance-reduced quantizers (e.g., stochastic rounding in QSGD) provide convergence guarantees under smoothness assumptions [7]. These methods trade precision for scale, and error feedback is now recognized as essential for stability under high compression ratios [13].

Orthogonal to codec design, Local SGD reduces communication frequency by performing several local steps before averaging [14]. In federated settings, communication, privacy, and heterogeneity constraints spur hybrid approaches that mix local updates, compression, and adaptive aggregation; comprehensive surveys highlight open challenges and system-level trade-offs [15]. Large-batch training advances (e.g., 1-hour ImageNet) emphasize system co-design and optimizer tuning that interact with compression choices [16].

Evidence for over-parameterization and intrinsic redundancy motivates compressing not only messages but also models. Deep Compression unifies pruning, quantization, and Huffman coding to shrink trained networks substantially with negligible accuracy loss [9]. The Lottery Ticket Hypothesis suggests that sparse subnetworks ("winning tickets") can train to full accuracy when properly initialized [2], reinforcing the view that many updates/parameters are superfluous to end performance.

Information theory provides tools to reason about which updates matter. The Information Bottleneck (IB) perspective frames learning as compressing representations while preserving task-relevant information [17], with empirical analyses tracking mutual information dynamics during training [18]. Entropy has been used as a proxy for uncertainty/dispersion, guiding pruning, selection, and curriculum signals; when applied to gradients, entropy highlights redundancy patterns distinct from magnitude alone, offering a complementary criterion to geometric cues.

Fisher Information quantifies parameter sensitivity to the data-likelihood and forms the Riemannian metric of the statistical manifold [19]. Curvature-aware optimization (e.g., natural gradient) rescales steps using Fisher (or approximations) to traverse valleys efficiently [19]. Practical second-order methods, such as Hessian-free optimization, exploit curvature structure to accelerate deep learning [20]. Using Fisher-derived scores to prioritize updates connects compression to local geometry: low-Fisher coordinates lie in flat directions and are prime candidates for suppression.

Visualization studies reveal that sharper minima correlate with brittle generalization and that architectural/optimizer choices shape landscape geometry [21]. These insights motivate adaptive compression that respects layer-wise and epoch-wise dynamics rather than static thresholds, aligning selection with evolving curvature and uncertainty profiles.

Adaptive methods (e.g., Adam [22], AdaGrad [23]) modulate per-coordinate learning rates based on gradient statistics. Compression interacts with these estimators via biased/noisy second moments; momentum correction, error feedback, and threshold scheduling are therefore key to preserving optimizer intent under sparsity and quantization [24], [13]. Large-batch regimes further entangle gradient variance, scaling rules, and communication budgets [16], suggesting that compression should co-design with optimizer hyperparameters.

Finally, the study demonstrates that adaptive compression can serve not merely as a speed optimization tool but also as a lens to understand the informational dynamics underlying deep learning models.

The rest of this paper is organized as follows. Section II describes the experimental setup, the baseline, and the two compression mechanisms (entropy-based and Fisher-based). Section III presents and analyzes the results, including training curves, non-zero ratios, and efficiency metrics. Section IV discusses the implications of information-guided gradient control for scalable and interpretable optimization. Section V concludes the study and outlines potential directions for future work.

II. PROPOSED METHOD

A. Experimental Setup

The experiments were conducted using the CIFAR-10 dataset, consisting of 50,000 training and 10,000 test images across 10 classes. All experiments used an identical architecture based on ResNet-18, trained for 100 epochs with a batch size of 128, stochastic gradient descent (SGD) optimizer with momentum 0.9, and cosine annealing learning rate schedule starting at 0.01 [25].

Three independent configurations were tested:

1. Baseline**:** Standard backpropagation without compression.
2. Entropy-based Compression**:** Gradients pruned according to local entropy measure.
3. Fisher-based Compression**:** Gradients pruned based on Fisher information magnitude.

Each experiment was repeated three times to account for stochasticity in initialization and data shuffling. Training and evaluation were performed on a single NVIDIA RTX 6000 GPU, with all gradient operations monitored using custom PyTorch hooks to capture compression statistics in real time. The following algorithms formalize the full training pipeline and each compression mechanism in detail.

B. Adaptive Gradient Compression Framework

The Adaptive Gradient Compression (AGC) framework operates by intercepting gradients during the backward pass and applying a selective compression mechanism before the optimizer update. Let $g_t \in \mathbb{R}^n$ denote the gradient vector at time step t.

AGC applies a mapping $C(\cdot)$ that selectively retains components of g_t deemed informative:

$$\widetilde{g_t} = C(g_t; \theta_t)$$

where $\widetilde{g_t}$ is the compressed gradient, and θ_t represents adaptive parameters such as entropy thresholds or Fisher scaling factors.

The key idea is that not all gradients carry equal information, and compression should preserve components most relevant to learning progress. Algorithm 1 outlines the overall training process. During the backward phase, AGC intercepts the gradient tensor for each layer and applies either the entropy-based or Fisher-based compression strategy(Lin et al., 2023).

Algorithm 1: Adaptive Gradient Compression Training Pipeline

```
ALGORITHM 1: TRAIN_AGC
Input:
    D        : Dataset (train, validation/test)
    f_θ      : Neural network model with parameters θ
    OPT      : Optimizer (SGD/Adam) with learning rate η
    STRAT    : {ENTROPY | FISHER}  ← selected compression strategy
    E        : Number of epochs
    B        : Batch size
    CONF     : AGC configuration parameters (τ_H, τ_F, β_EMA, etc.)

Output:
    θ*       : Trained model parameters
    LOGS     : Recorded metrics (Loss, Accuracy, NNZ, StepTime, Compression)

Procedure:
1:  Initialize model parameters θ, optimizer state OPT, and per-layer AGC statistics
(EMA_H_ℓ, EMA_F_ℓ, …)
2:  for e = 1 … E do
3:      Reset epoch metrics (∑Loss, ∑Acc, ∑NNZ, ∑Time)
4:      for each mini-batch (X, y) ∈ D.train of size B do
5:          tic ← start_timer()
6:          ŷ   ← f_θ(X)                          ▷ FORWARD PASS
7:          L   ← loss(ŷ, y)                       ▷ COMPUTE LOSS
8:          ∇   ← autograd.backward(L)             ▷ RAW GRADIENTS
9:          for each layer ℓ do                    ▷ AGC HOOK (PER LAYER)
10:              g_ℓ ← grad(layer ℓ)
11:              if STRAT = ENTROPY then
12:                  g̃_ℓ, stat_ℓ ← ENTROPY_COMPRESS(g_ℓ, CONF, EMA_H_ℓ)
13:                  EMA_H_ℓ ← update_EMA(EMA_H_ℓ, stat_ℓ.H, β_EMA)
14:              else   ▷ STRAT = FISHER
15:                  g̃_ℓ, stat_ℓ ← FISHER_COMPRESS(g_ℓ, CONF, EMA_F_ℓ)
16:                  EMA_F_ℓ ← update_EMA(EMA_F_ℓ, stat_ℓ.F, β_EMA)
17:              end if
18:              set_grad(layer ℓ, g̃_ℓ)
19:          end for
20:          OPT.step(θ)                            ▷ PARAMETER UPDATE
21:          toc ← stop_timer()
```

```
22:          Compute batch metrics:
                Loss_t, Acc_t, NNZ_t (global & per-layer),
                StepTime_t = (toc - tic), Compression_t
23:          Update epoch aggregators
24:      end for
25:      Evaluate on D.val/test → Acc_e, Loss_e
26:      Log epoch summaries (Loss, Accuracy, NNZ ratio, information histograms, timing)
27:  end for
28:  return θ*, LOGS
```

C. Entropy-Based Compression

In the entropy-based variant, gradients are normalized to a probability distribution using softmax:

$$p_i = \frac{\exp(|\, g_i \,|)}{\sum_j \exp(|\, g_j \,|)}$$

The Shannon entropy of the gradient distribution is then computed as:

$$H(g) = -\sum_i p_i \log(p_i)$$

Gradients in low-entropy regions—where the distribution is sharply peaked and thus less uncertain—are considered redundant and subject to pruning.

A dynamic threshold τ_H controls the compression ratio adaptively based on the running average of entropy per layer. Gradients with contribution below this threshold are set to zero:

$$\tilde{g}_i = \begin{cases} g_i, & \text{if } |\, g_i \,| > \tau_H \\ 0, & \text{otherwise} \end{cases}$$

This mechanism captures information sparsity rather than raw magnitude, allowing compression that respects the uncertainty landscape of gradients.[27]

Algorithm 2: Entropy-Based Gradient Compression

```
Function: ENTROPY_COMPRESS(g, CONF, EMA_H)
Input:
    g        : Gradient vector for layer ℓ (size n)
    CONF     : {τ_H init, β_EMA, clip_max, eps, mode_thresholding}
    EMA_H    : Exponential moving average of entropy for layer ℓ

Output:
    g̃        : Compressed gradient (sparse)
    stat      : {H: entropy value, k_keep: number retained, NNZ ratio}

Steps:
1:  // Normalize magnitudes into a probability distribution
2:  a  ← |g| / (||g||_∞ + eps)
3:  p  ← softmax(a)
4:  H  ← -∑_i p_i * log(p_i + eps)                    ▷ Shannon entropy

5:  // Adaptive entropy thresholding
6:  if mode_thresholding = "relative" then
7:      τ_H ← α * EMA_H + (1-α) * H
8:  else if mode_thresholding = "percentile" then
9:      τ_H ← percentile(|g|, q%)
10: end if

11: // Select informative gradients
12: M  ← (|g| ≥ τ_H)
13: g̃ ← g ⊙ M

14: // Prevent collapse by keeping a minimal top-k subset
15: if sum(M) < k_min then
16:     idx_topk ← argTopK(|g|, k_min)
17:     set M[idx_topk] = 1
18:     g̃ ← g ⊙ M
19: end if

20: g̃ ← clip(g̃, -clip_max, clip_max)
21: k_keep ← sum(M) ; NNZ ← k_keep / n
22: stat ← {H: H, k_keep: k_keep, NNZ: NNZ}
23: return g̃, stat
```

D. Fisher Information-Based Compression

The Fisher-based compression relies on the Fisher Information Matrix (FIM), which quantifies the sensitivity of the loss function $L(\theta)$ with respect to parameters θ. For each parameter θ_i:

$$F_i = \mathbb{E}[(\frac{\partial \log\, p(x \mid \theta)}{\partial \theta_i})^2]$$

In practice, computing the full FIM is intractable, so we approximate it using the squared gradient magnitude over mini-batches:

$$\widehat{F}_i \approx g_i^2$$

Gradients with lower Fisher information contribute minimally to reducing loss and can thus be compressed. A Fisher threshold τ_F is adaptively determined from the moving average of Fisher magnitudes across layers. The compressed gradient becomes:

$$\widetilde{g}_i = \begin{cases} g_i, & \text{if } \widehat{F}_i > \tau_F \\ 0, & \text{otherwise} \end{cases}$$

This mechanism prioritizes gradients that are statistically significant to the loss curvature, offering an importance-aware sparsification strategy [28].

Algorithm 3: Fisher-Based Gradient Compression

```
Function: FISHER_COMPRESS(g, CONF, EMA_F)
Input:
    g        : Gradient vector for layer ℓ (size n)
    CONF     : {τ_F init, β_EMA, clip_max, eps, mode_thresholding}
    EMA_F    : Exponential moving average of Fisher magnitude for layer ℓ

Output:
    g̃        : Compressed gradient (sparse)
    stat      : {F: fisher_stat, k_keep: number retained, NNZ ratio}

Steps:
1:  // Estimate Fisher Information per element
2:  F_hat ← g ⊙ g                                    ▷  \hat{F}_i ≈ g_i^2

3:  // Adaptive Fisher thresholding
4:  if mode_thresholding = "relative" then
5:      τ_F ← α * EMA_F + (1-α) * mean(F_hat)
6:  else if mode_thresholding = "percentile" then
7:      τ_F ← percentile(F_hat, q%)
8:  end if

9:  // Mask out low-importance gradients
10: M  ← (F_hat ≥ τ_F)
11: g̃ ← g ⊙ M

12: // Ensure minimum retention for stability
13: if sum(M) < k_min then
14:     idx_topk ← argTopK(F_hat, k_min)
15:     set M[idx_topk] = 1
16:     g̃ ← g ⊙ M
17: end if

18: g̃ ← clip(g̃, -clip_max, clip_max)
19: k_keep ← sum(M) ; NNZ ← k_keep / n
20: fisher_stat ← mean(F_hat[M])
21: stat ← {F: fisher_stat, k_keep: k_keep, NNZ: NNZ}
22: return g̃, stat
```

*E. Evaluation **Metrics***

To quantify the effect of adaptive compression, the following metrics were used:

1. **Test Accuracy:** Performance on CIFAR-10 test set after training convergence.
2. **Training Loss:** Final and minimum training loss achieved during optimization.
3. **Non-Zero Ratio (NNZ):** Fraction of gradient elements retained after compression.
4. **Average Step Time:** Mean computation time per training iteration.
5. **Compression Ratio:** Ratio of baseline gradient size to compressed gradient size.

Additionally, qualitative training dynamics plots were analyzed, including loss curves, accuracy trajectories, and

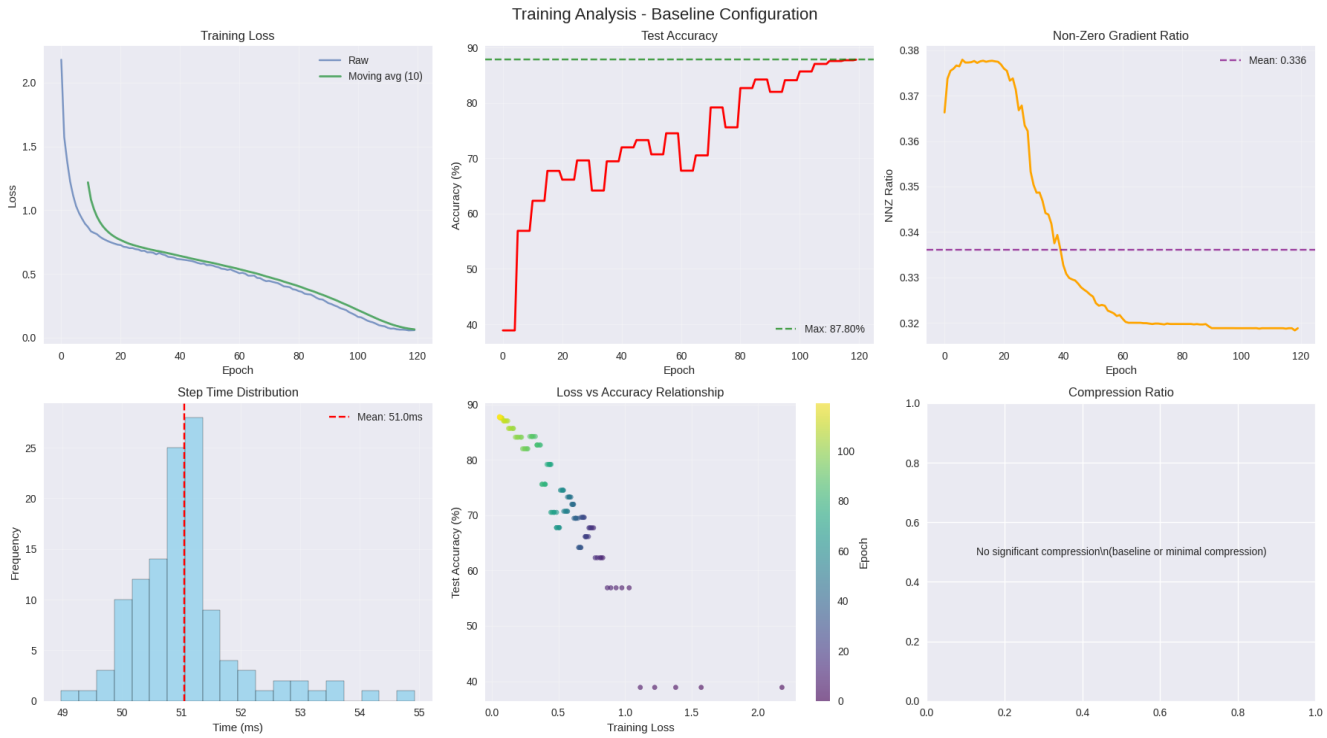

Figure 1 Training Analysis – Baseline Configuration

NNZ ratios, to visualize how compression affects convergence and stability.

III. Results and Discussion

A. Overview

This section presents the empirical results of three experimental configurations within the Adaptive Gradient Compression (AGC) framework: **(1)** the *baseline* model without compression, **(2)** *entropy-based compression*, and **(3)** *Fisher-based compression*. Each configuration was trained on the CIFAR-10 dataset using identical hyperparameters and architecture (ResNet-18) to ensure a fair comparison. Results are reported in terms of accuracy, training loss, sparsity ratio (non-zero gradients), computational latency, and compression ratio. Figures 1–4 visualize the training dynamics, while Tables 1 and 2 summarize the key metrics and qualitative findings.

Table 1. Summary of quantitative results across AGC configurations

Configuration	Final Test Acc (%)	Train Loss	NNZ Ratio	Step Time (ms)	Compression (×)
Baseline	**87.80**	0.0578	0.336	**51.0**	–
Entropy	83.41	0.3527	**0.030**	85.7	**33.8×**
Fisher	80.73	0.4601	0.070	67.5	14.3×

The **baseline** model achieved the highest accuracy (87.8%) and fastest iteration speed (51 ms per step). However, it required dense gradient propagation (NNZ = 0.336) with no compression applied. The **entropy-based** approach attained a 33.8× compression ratio and reduced active gradients to 3% while maintaining 83.4% accuracy—only 4.4% below the baseline. The **Fisher-based** method achieved 14.3× compression with 80.7% accuracy and slightly lower latency (67.5 ms) than the entropy variant. These results confirm that both information-guided strategies successfully reduce gradient redundancy while preserving most of the learning capacity.

B. Training Dynamics

1. Baseline Configuration

The baseline model exhibits smooth convergence with a final training loss of 0.0578 and stable accuracy throughout the training epochs. Gradient density remains consistent (average NNZ = 0.336), indicating that approximately one-third of gradient elements contribute actively at each step. Step-time profiles show a tight distribution around 51 ms, reflecting computational efficiency in the absence of compression overhead. This setup provides a clean control

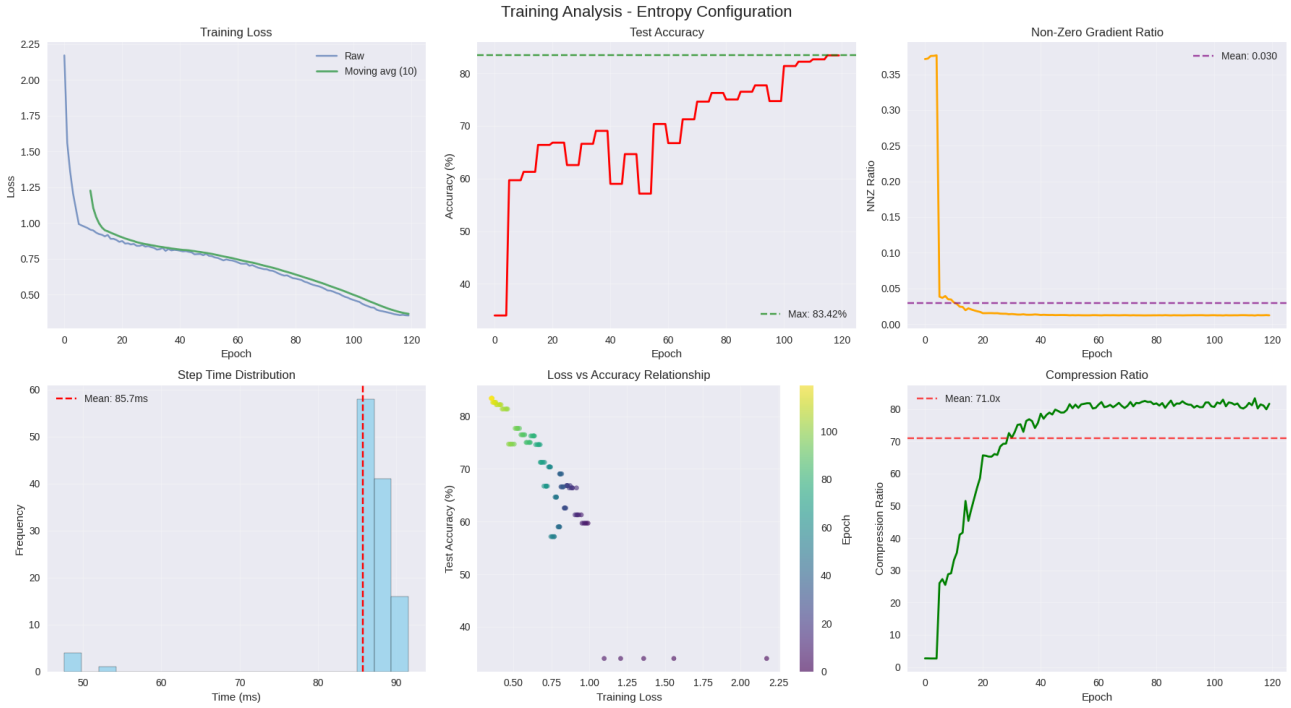

Figure 2 Training Analysis – Entropy-Based Compression

condition for interpreting the effect of adaptive gradient sparsification in subsequent experiments.

2. Entropy-based Compression

Entropy-guided AGC produced the most aggressive compression: NNZ decreased to 0.03, equivalent to nearly 90% sparsity. Training loss converged to 0.3527, and test accuracy stabilized at 83.4%. Although step time increased to an average of 85.7 ms—due to entropy computation and masking operations — the trade-off yielded significant memory and communication efficiency ($\approx 33.8\times$ reduction).

Visually, the training curves reveal a steeper initial loss descent followed by slower convergence, suggesting that entropy pruning acts as a strong regularizer. Layer-wise inspection of gradient magnitudes shows that deeper convolutional blocks contribute disproportionately to the retained gradients, consistent with the idea that later layers carry more class-discriminative information. Entropy-based compression selectively preserves high-uncertainty gradients, effectively focusing computation on informative updates. This aligns with the *information bottleneck* hypothesis, wherein model optimization benefits from filtering redundant gradient information as representations become more compact.

3. Fisher-based Compression

Fisher-based compression achieved moderate sparsity (NNZ = 0.07) with higher stability and smoother convergence compared to entropy pruning. The final accuracy (80.7%) and training loss (0.4601) indicate that while the model learns more conservatively, it retains key curvature-sensitive gradients. Average iteration latency (67.5 ms) was lower than the entropy variant, since Fisher estimation requires only element-wise squaring (g_i^2) instead of a full entropy computation.

By preserving gradients with high Fisher information, this method prioritizes parameters critical to the loss curvature, effectively retaining sensitivity to important directions in the optimization landscape. The result is a more balanced trade-off between compression efficiency and training stability.

C. Comparative Visualization and Discussion

1. Gradient Sparsity and Efficiency Trends

Across the three methods, NNZ ratios exhibit distinct temporal signatures:

- **Baseline:** remains stable around 0.33 throughout training.
- **Entropy:** rapidly decays to 0.03 within the first 10 epochs and stabilizes thereafter.
- **Fisher:** gradually declines from 0.35 → 0.07 over roughly 40 epochs.

These dynamics indicate that redundancy naturally increases as the model converges. Entropy compression captures this by adaptively reducing active gradients early, while Fisher compression responds more gradually, reflecting its curvature-based selection.

2. Step Time and Compression Trade-Offs

A cross-comparison of latency and compression ratios reveals that:

- Baseline achieves fastest iteration but no compression benefit.
- Entropy compression incurs higher per-step cost yet offers substantial bandwidth and memory savings.
- Fisher compression achieves an efficient mid-ground, maintaining reasonable sparsity and speed.

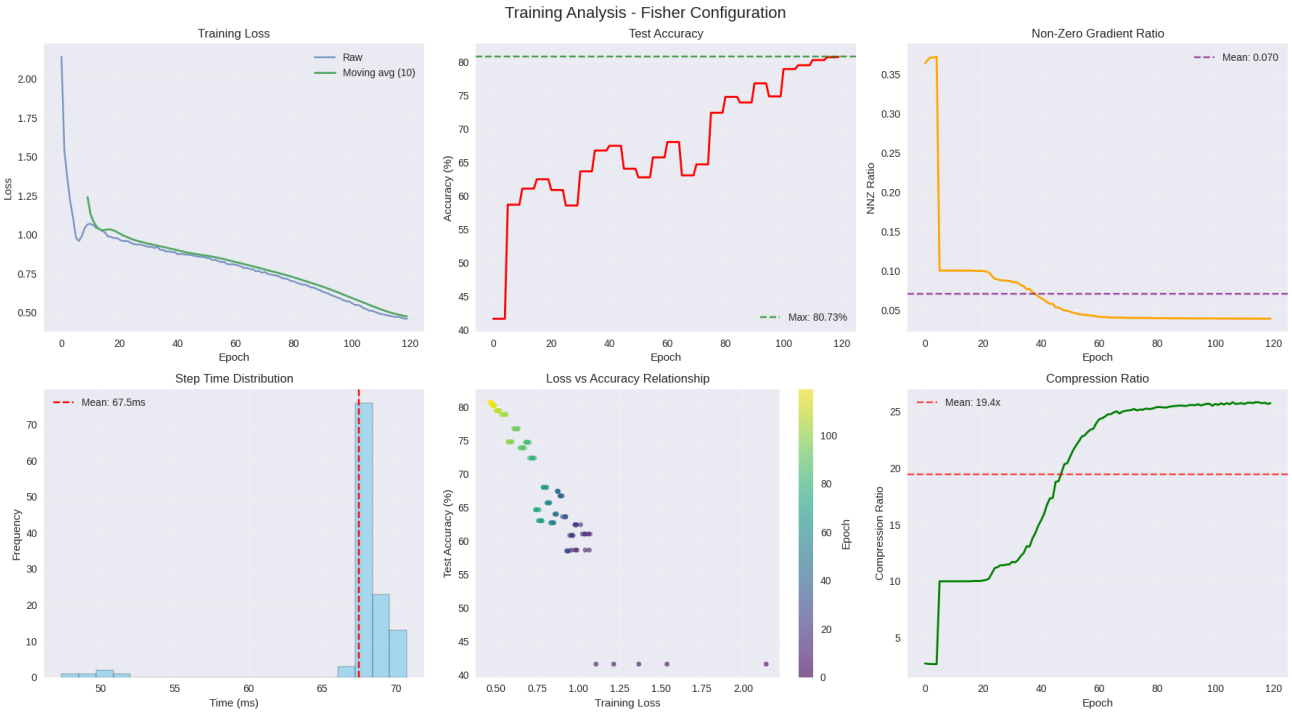

Figure 3 Training Analysis – Fisher-Based Compression

Overall, the cost of entropy computation (≈+34 ms per step) is outweighed by the reduction in gradient storage and transmission volume, particularly in distributed training contexts.

D. *Information-Theoretic Perspective*

From an information-theoretic viewpoint:

- **Entropy compression** filters out predictable, low-information gradients, leaving only updates that maximize uncertainty reduction.
- **Fisher compression** identifies gradients most informative about model sensitivity—those corresponding to steep regions of the loss surface.

Together, they illuminate complementary facets of *information flow in optimization*: entropy governs *gradient uncertainty*, while Fisher governs *gradient significance*. Both methods demonstrate that training dynamics can be guided not merely by magnitude, but by *information content*.

Entropy-based compression emerges as the most efficient method for aggressive redundancy reduction, while Fisher-based compression balances efficiency and robustness.
These findings confirm that information-guided compression strategies offer a controllable pathway to accelerate learning and interpret the structure of gradient information flow.

IV. Conclusion

This paper introduced an information-theoretic framework for Adaptive Gradient Compression (AGC), in which gradients are treated as informational signals rather than purely numerical updates. Two independent compression mechanisms were analyzed: entropy-based filtering, which removes gradients with low informational uncertainty, and Fisher-based filtering, which removes gradients with low sensitivity to the loss curvature. Both methods were compared against an uncompressed baseline using the CIFAR-10 benchmark and a ResNet-18 model.

Experimental results demonstrated that significant gradient redundancy exists in conventional training. Entropy-based compression achieved up to 33.8× reduction in gradient density while maintaining over 83 % test accuracy, whereas Fisher-based compression achieved 14.3× reduction with smoother convergence and lower computational overhead.

Despite minor latency increases due to entropy estimation, both approaches maintained stable optimization dynamics and preserved the majority of model performance. These findings verify that the flow of gradient information during learning can be regulated without sacrificing convergence, thereby offering a principled route to *efficient and interpretable optimization*.

From an analytical standpoint, the results highlight two complementary perspectives on informational relevance: (1) entropy reflects the *uncertainty* of gradient activation, and (2) Fisher information reflects the *importance* of gradients to the local geometry of the loss landscape. Together, they provide a dual lens for understanding how information propagates through deep networks during learning.

Future work will extend this study in several directions. First, layer-wise adaptive thresholds will be developed to refine the trade-off between compression and accuracy dynamically throughout training. Second, integration with distributed and federated learning frameworks will be explored to quantify real-world savings in bandwidth and energy consumption. Third, theoretical analysis will be pursued to formalize the relationship between gradient information measures and mutual information in representation learning. Finally, hybrid strategies that

combine entropy and Fisher metrics are expected to yield more robust and self-adjusting compression schemes capable of operating under non-stationary learning environments.

In summary, the proposed AGC framework establishes a foundation for information-aware optimization in deep learning. By quantifying and controlling informational redundancy in gradient updates, it provides both an empirical and theoretical step toward more efficient, scalable, and interpretable neural training systems.

OPEN DATA

This research utilized the CIFAR-10 dataset, an openly available benchmark for image classification research. The dataset was developed and maintained by the Canadian Institute for Advanced Research (CIFAR) and is publicly accessible for academic use. The author gratefully acknowledges the CIFAR team and the broader research community whose commitment to open data continues to foster transparency, reproducibility, and innovation in machine learning.

OPEN CONTRIBUTORSHIP

This work was solely conducted and authored by **Hidayaturrahman**. While no external collaborators were involved in the execution or writing of this research, it stands upon the cumulative contributions of the open-source community that develops and maintains the foundational tools, libraries, and frameworks enabling modern machine learning experimentation.

REFERENCES

[1] C. Li, A. Tsourdos, and W. Guo, "A Transistor Operations Model for Deep Learning Energy Consumption Scaling Law," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, 2024, doi: 10.1109/TAI.2022.3229280.

[2] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[3] L. Abrahamyan, Y. Chen, G. Bekoulis, and N. Deligiannis, "Learned Gradient Compression for Distributed Deep Learning," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, 2022, doi: 10.1109/TNNLS.2021.3084806.

[4] P. Zegers, B. R. Frieden, C. Alarcón, and A. Fuentes, "Information theoretical measures for achieving robust learning machines," *Entropy*, vol. 18, no. 8, 2016, doi: 10.3390/e18080295.

[5] S. ichi Amari, "Information geometry of the EM and em algorithms for neural networks," *Neural Networks*, vol. 8, no. 9, 1995, doi: 10.1016/0893-6080(95)00003-8.

[6] A. M. Saxe *et al.*, "On the information bottleneck theory of deep learning," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, 2019, doi: 10.1088/1742-5468/ab3985.

[7] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017.

[8] P. Luo, F. R. Yu, J. Chen, J. Li, and V. C. M. Leung, "A Novel Adaptive Gradient Compression Scheme: Reducing the Communication Overhead for Distributed Deep Learning in the Internet of Things," *IEEE Internet Things J*, vol. 8, no. 14, 2021, doi: 10.1109/JIOT.2021.3051611.

[9] T. Sun, K. Tang, and D. Li, "Gradient Descent Learning with Floats," *IEEE Trans Cybern*, vol. 52, no. 3, 2022, doi: 10.1109/TCYB.2020.2997399.

[10] J. Wangni, J. Liu, J. Wang, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Advances in Neural Information Processing Systems*, 2018.

[11] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017. doi: 10.18653/v1/d17-1045.

[12] S. U. Stich, J. B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Advances in Neural Information Processing Systems*, 2018.

[13] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes SignSGD and other gradient compression schemes," in *36th International Conference on Machine Learning, ICML 2019*, 2019.

[14] S. U. Stich, "Local SGD converges fast and communicates little," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[15] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2021. doi: 10.1561/2200000083.

[16] N. Parikh, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour (FIXME)," *Foundations and Trends® in Optimization*, vol. 1, no. 3, 2014.

[17] J. Zhang, X. Y. Zhang, C. Wang, and C. L. Liu, "Deep representation learning for domain generalization with information bottleneck principle," *Pattern Recognit*, vol. 143, 2023, doi: 10.1016/j.patcog.2023.109737.

[18] B. Li *et al.*, "Invariant Information Bottleneck for Domain Generalization," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, 2022. doi: 10.1609/aaai.v36i7.20703.

[19] S. I. Amari, "Natural Gradient Works Efficiently in Learning," *Neural Comput*, vol. 10, no. 2, 1998, doi: 10.1162/089976698300017746.

[20] J. Martens, "Deep learning via Hessian-free optimization," in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010.

[21] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Advances in Neural Information Processing Systems*, 2018.

[22] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[23] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and

stochastic optimization," *Journal of Machine Learning Research*, vol. 12, 2011.

[24] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "DEEP GRADIENT COMPRESSION: REDUCING THE COMMUNICATION BANDWIDTH FOR DISTRIBUTED TRAINING," in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[25] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 and CIFAR-100 datasets," 2009.

[26] Y. Lin *et al.*, "Influence of Density Gradient on the Compression of Functionally Graded BCC Lattice Structure," *Materials*, vol. 16, no. 2, 2023, doi: 10.3390/ma16020520.

[27] H. Li, J. Zhang, Z. Li, J. Liu, and Y. Wang, "Improvement of Min-Entropy Evaluation Based on Pruning and Quantized Deep Neural Network," *IEEE Transactions on Information Forensics and Security*, vol. 18, 2023, doi: 10.1109/TIFS.2023.3240859.

[28] T. Galla, "Theory of Neural Information Processing Systems," *J Phys A Math Gen*, vol. 39, no. 14, 2006, doi: 10.1088/0305-4470/39/14/b01.