

Sparse And Efficient Models for Low Power Devices

Renjusha P R¹, Aparna A²

¹Postgraduate Student, Department of Computer Application, Nehru College of Engineering and Research Centre, Pampady, Kerala, India.

²Assistant Professor, Department of Computer Application, , Nehru College of Engineering and Research Centre, Pampady, Kerala, India.

Abstract

Parsimony, including concepts like sparsity and low-rank structure, has proven effective for modeling data in many machine learning and signal processing applications. Traditionally, these approaches depend on iterative optimization methods that minimize objective functions with parsimony-related penalties. However, these methods are inherently sequential and their complexity depends on the data, which limits their practicality in real-time or large-scale scenarios. Another drawback is that incorporating them into discriminative learning frameworks is often challenging. In this work, the focus shifts from the model itself to the pursuit algorithm, introducing a process-centric perspective. Instead of relying on iterative optimization, we design deterministic, fixed-complexity architectures inspired by traditional sparse and low-rank methods. These architectures are derived from approximations of proximal optimization algorithms and are capable of producing accurate parsimonious representations at a fraction of the computational cost of standard methods. We also demonstrate that, under suitable training conditions, this approach can naturally extend to discriminative tasks. Experimental results on several challenging image and audio processing problems show state-of-the-art performance, achieving speed improvements of several orders of magnitude compared to exact optimization techniques.

1. Introduction

Simplicity, which favours straightforward explanations over more intricate ones, stands out as one of the most natural principles commonly utilized in modelling the natural world. Research over the last twenty years has demonstrated the effectiveness of simple representations across a wide range of scientific disciplines. Among the simplest forms of parsimonious models is the concept of sparsity, which posits that a signal consists of numerous coefficients that are near or equal to zero when expressed in a certain domain, typically referred to as a dictionary. The quest for sparse representations has been shown to be achievable using methods from convex optimization, particularly using l_1 norm minimization. Subsequent studies introduced by others have developed efficient computational methods for learning and adapting dictionaries. Sparse modelling is central to contemporary techniques in image enhancement, including denoising, demosaicing, inpainting, and super-resolution, among others. Since various data types are poorly represented by the element-wise sparsity model and the associated l_1 norm, more complex structured sparse models have emerged, where non-zero elements are interrelated and organized into groups or hierarchies of groups. These structured models have proven beneficial in analysing functional MRI and genetic data, for instance. In the context of matrix-valued data, the rank serves as a natural measure of complexity,

which also leads to a concept of parsimony. A recent collection of studies has clarified the elegant connection between sparsity and low-rank representations, demonstrating that it is possible to achieve rank minimization through convex optimization. The integration of low-rank and sparse models has opened up new, robust alternatives to principal component analysis (RPCA) and nonnegative matrix factorization (RNMF), as well as tackled complex matrix completion challenges. RPCA has also proven beneficial in significant applications, including face recognition and modeling, background modeling, and audio source separation. Another pertinent low-rank modeling approach is non-negative matrix factorization (NMF), where the input vectors are expressed as non-negative linear combinations of a non-negative under complete dictionary. NMF has achieved notable success in fields such as object recognition and audio processing.

2. Literature Review

Recent studies increasingly emphasize the design of sparse and efficient models to support artificial intelligence on low-power and edge devices. Jo and Kim [1] provide a comparative analysis of lightweight sparse autoencoder-based classifiers for edge networks, showing that incorporating sparsity can greatly reduce computational load and energy usage while maintaining strong classification performance. Their findings suggest that well-structured sparse architectures can surpass deeper models in resource-limited settings, highlighting the value of efficiency-focused design rather than relying solely on network depth.

Model Compression for Edge Deployment

Model compression has become essential for deploying advanced models on edge platforms. Tanvir et al. [2] introduce an extreme compression framework for edge vision–language models that integrates sparse temporal token fusion with adaptive neural compression. By leveraging redundancy across spatial and temporal features, their method significantly lowers memory usage and inference time with minimal performance loss. This demonstrates the importance of structured sparsity and adaptive compression in achieving an effective balance between accuracy and efficiency in multimodal edge applications.

Pruning for Reducing Model Complexity

Pruning methods are widely recognized for simplifying neural networks and enabling efficient edge deployment. Crespí-Castañer et al. [3] examine both structured and unstructured pruning techniques applied to dense models, showing that a large number of parameters can be removed without notable accuracy degradation when guided by suitable importance metrics. Their work also indicates that pruning enhances energy efficiency and speeds up inference, making it particularly advantageous for real-time edge scenarios.

Optimization Techniques for Edge Inference

Broad survey studies such as [4] and [5] review key optimization strategies for edge inference, including sparsity, pruning, quantization, and hardware-aware design. These surveys stress the need for predictable runtime and low-power operation while addressing challenges like limited memory, restricted computation, and heterogeneous edge hardware. Overall, they reinforce the importance of integrating algorithmic sparsity with system-level optimization to enable efficient and scalable edge AI deployment.

3. Methodology

The methodology focuses on building sparse and efficient models by replacing traditional iterative optimization with fixed-complexity, learnable inference processes suitable for low-power and real-time environments. It follows a process-centric approach that learns the representation mechanism instead of

solving optimization problems during inference, ensuring predictable runtime and reduced computational cost. Input data are represented using sparse or low-rank structures based on a learned dictionary, with preprocessing steps such as normalization, feature extraction, or vectorization. Dictionaries are initialized using standard methods and refined during training to improve representation quality. The core method designs a learnable pursuit process inspired by proximal algorithms, where a fixed number of optimization steps are unrolled into a feedforward network with learnable parameters, resulting in a deterministic encoder with fixed depth and efficient performance. The encoder is trained using stochastic gradient descent to minimize a loss function based on the task. In unsupervised learning, the loss focuses on reconstruction error, while in supervised settings it includes objectives like classification accuracy or source separation. Gradients are computed using backpropagation through the unrolled architecture. The methodology supports online learning, allowing incremental updates with streaming or mini-batch data. It also handles transformed or misaligned data by jointly learning differentiable transformation parameters. Model performance is evaluated using reconstruction error, sparsity level, and classification accuracy, while efficiency is measured through inference time, memory usage, and runtime consistency. Results are compared with traditional iterative methods to show improved speed and suitability for low-power devices.

Overall, this methodology integrates principled sparse and low-rank modelling with learnable inference processes to achieve efficient, scalable, and practical solutions. By combining model-based structure with data-driven learning, the proposed approach provides a robust framework for deploying parsimonious models on low-power and real-time systems.

4. Parsimonious Models and Proximal Methods

Parsimonious models are founded on the idea that complex data can often be represented using a small number of meaningful components. Rather than relying on dense representations, these models seek compact structures that preserve essential information, leading to improved interpretability, robustness to noise, and better generalization. Given a data matrix $X \in \mathbb{R}^{m \times n}$, parsimonious modeling aims to obtain a compact representation by solving an optimization problem over a representation matrix $Z \in \mathbb{R}^{q \times n}$, either with a fixed dictionary $D \in \mathbb{R}^{m \times q}$ (parsimonious coding) or by jointly learning both Z and D (parsimonious modeling). While coding assumes a predefined dictionary and is computationally simpler, joint learning is more challenging due to its non-convex nature. In large-scale or streaming scenarios, online approaches update representations and dictionaries incrementally, enabling scalable and memory-efficient learning. Structured sparsity extends basic sparse modeling by enforcing patterns among nonzero coefficients, such as groups or hierarchies, rather than treating them independently. Classical sparse coding commonly employs ℓ_1 -norm regularization (Lasso) to promote sparsity. Low-rank modeling represents another form of parsimony, where data are approximated using a limited number of latent components. Principal Component Analysis (PCA) decomposes data into low-rank structure and noise but is sensitive to outliers. Robust PCA (RPCA) addresses this limitation by separating the data into low-rank and sparse corruption components using nuclear norm and ℓ_1 -norm regularization. Similarly, Non-negative Matrix Factorization (NMF) factorizes a non-negative data matrix into low-rank non-negative components and can be extended with sparse terms to improve robustness against outliers.

Proximal methods provide an efficient optimization framework for solving sparse and structured problems involving non-smooth regularization. These iterative algorithms alternate between a gradient step on the data-fidelity term and a proximal operation that enforces structural constraints such as sparsity or non-

negativity. For ℓ_1 -regularized problems, the proximal operator reduces to soft-thresholding, as in the Iterative Shrinkage-Thresholding Algorithm (ISTA), with accelerated variants achieving faster convergence. Proximal techniques naturally extend to structured sparsity, RPCA, and robust NMF, offering strong convergence guarantees and computational efficiency for large-scale and online applications.

5. Learnable Pursuit Processes

The general parsimonious modeling problem can be reformulated as a supervised learning task. Instead of solving an optimization problem separately for each input, the objective is to learn a mapping that directly generates a compact representation. This is typically implemented using an encoder–decoder architecture, where the encoder transforms the input data into a low-dimensional representation, and the decoder reconstructs the original signal from this representation. The reconstruction is encouraged to approximate the identity mapping, while regularization promotes properties such as sparsity and structural constraints. In learnable pursuit processes, both the encoder and decoder are defined as deterministic parametric functions. The encoder operates as a fixed-complexity mapping with trainable parameters, enabling representations to be computed through a single forward pass rather than iterative optimization. The decoder is often kept linear to maintain simplicity and interpretability. This framework involves two key challenges: selecting an appropriate encoder architecture capable of approximating optimal pursuit solutions, and learning its parameters from training data to minimize reconstruction error while enforcing desired regularization. By replacing iterative algorithms with trained encoders, this approach achieves fast, predictable inference, making it particularly suitable for real-time and resource-constrained applications.

5.1 Process learning

Process learning focuses on learning an encoder that directly maps input data to a compact representation, avoiding the need to solve an optimization problem for each input. The encoder is chosen from a restricted family of functions that ensures differentiability, stability, and low computational complexity. The learning problem is solved using alternating minimization between the encoder and the dictionary. When the dictionary is fixed, the main objective is to learn the encoder by minimizing the empirical risk over training samples, which approximates the expected performance on unseen data. Optimization is performed using stochastic gradient descent (SGD) with mini batches, where encoder parameters are updated through backpropagation. This approach enables scalable training and fast inference, making process learning well suited for real-time and low-power applications.

5.2 Approximation Accuracy

The process training error can be decomposed into three parts: approximation error, estimation error, and optimization error. The approximation error measures how well the best pursuit process within a restricted function family can approximate the ideal unrestricted pursuit process, and it mainly depends on the choice of the encoder family. The estimation error arises from learning using a finite training set instead of the true data distribution and decreases as the training size increases. The optimization error is due to approximate minimization of the empirical risk and can be reduced by increasing the number of optimization iterations. As estimation and optimization errors become negligible, the overall performance is largely determined by the approximation capability of the chosen function family.

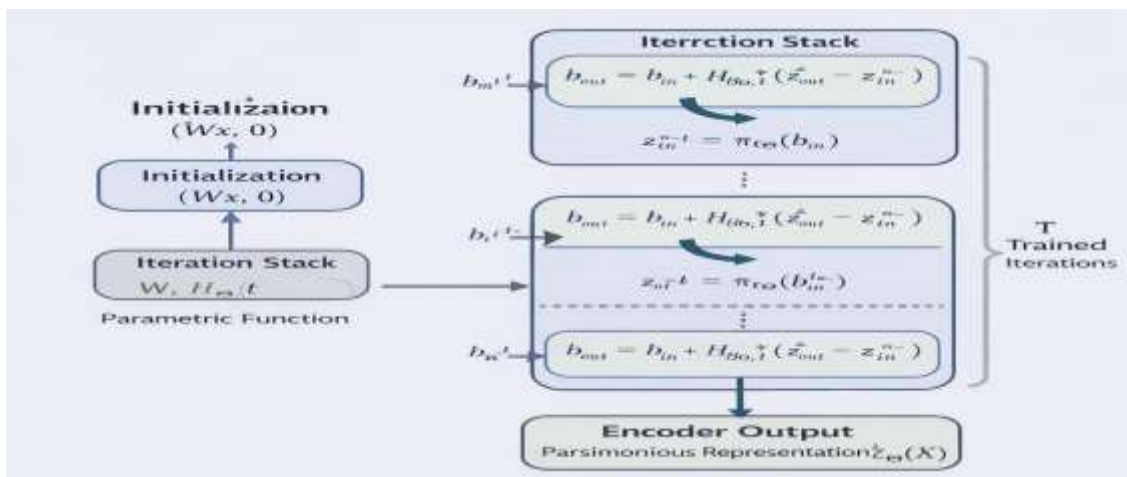
5.3 Process architecture

Proximal descent can be viewed as a parametric process in which each fixed-step iteration transforms the current state using a nonlinear proximal operator followed by a linear update. This transformation is

represented as a parametric function with learnable parameters, making the iteration itself trainable. A full proximal algorithm is obtained by stacking multiple such iterations, starting from a simple initialization based on the input data. Classical proximal algorithms naturally fit into this structure, while variations differ only in how state variables are updated at each step. Trainable pursuit processes are formed by truncating the iterative procedure to a fixed number of steps. All learnable parameters are grouped together, and only the final representation is used as the encoder output, enabling fast and predictable inference.

5.4 Approximation error vs. complexity trade-off

The analysis highlights a clear approximation error vs. complexity trade-off in learned pursuit processes. As the number of iterations T increases, the approximation error decreases at a rate proportional to $1/T$, meaning arbitrarily accurate solutions can be achieved in theory. However, this improvement comes with increased computational complexity, since the encoder’s cost grows linearly with T . While the theoretical bound is uniform and worst-case over all inputs, it is often loose in practice; better performance for likely inputs can be achieved by learning parameters tailored to the data distribution. These gains, however, require more parameters and higher offline training cost, underscoring the fundamental trade-off between accuracy, runtime complexity, and model size.



6. Training Regimes

Training regimes in parsimonious modelling range from unsupervised to supervised settings. In the unsupervised regime, encoder–decoder pairs are trained solely using data samples by minimizing a reconstruction-based empirical loss, enabling online learning with joint parameter and dictionary updates. Trainable pursuit processes bridge this gap by unrolling iterative optimization into a learnable architecture with fixed complexity and tuneable latency, offering an improved trade-off between computational cost and approximation error while better adapting to task-driven requirements.

6.1 Supervised Learning

Parsimonious models such as RPCA and RNMF provide effective first-order representations but remain limited because real-world data rarely adhere strictly to sparse or low-rank assumptions. Supervised learning addresses this mismatch by incorporating domain-specific knowledge directly into the training objective. In applications such as audio source separation, the availability of ground-truth targets allows encoder–decoder architectures to be trained to recover desired outputs rather than simply reconstructing the input.

In this framework, the encoder maps the input signal to latent components, while a linear decoder reconstructs the target signals. Although the encoder structure offers a strong initialization, supervised training enables better adaptation to real data, capturing structures beyond the original parsimonious assumptions. Consequently, supervised encoders consistently outperform unsupervised counterparts and traditional optimization-based methods. More generally, supervised learning defines the loss on task-specific outputs, enabling richer and more effective representations than approaches that merely approximate iterative algorithms.

6.2 Discriminative Learning

Discriminative learning extends parsimonious models from generative tasks to classification problems, where the objective is to distinguish between classes rather than reconstruct the input. In this setting, representations need not be invertible but are designed to capture discriminative and invariant features. Using a process-centric approach, multiple class-specific encoder–decoder pairs are trained simultaneously. Classification is performed by comparing reconstruction errors, with a discriminative loss that enforces low error for the correct class and higher errors for incorrect classes through a margin-based penalty. This framework demonstrates the effectiveness and flexibility of parsimonious models in supervised classification tasks.

6.3 Data Transformation

Parsimonious models typically assume well-aligned input data, an assumption often violated in practice, such as in face modelling. To overcome this limitation, geometric transformations are integrated directly into the modelling framework, enabling joint optimization of data alignment and representation learning. Each input sample is associated with transformation parameters that are optimized alongside the encoder and decoder. Since the encoder is differentiable, gradients with respect to these parameters can be efficiently computed using backpropagation. This approach supports online learning, allowing both model parameters and data alignment to be updated incrementally, and yields robust representations even under significant misalignment.

7. Application Domains

Parsimonious models play a central role in modern signal processing and machine learning by promoting compact, interpretable, and efficient representations of data. Due to their balance between expressive power and computational efficiency, parsimonious models have found widespread application across diverse domains, particularly in scenarios involving large-scale data, real-time processing, and limited computational resources.

7.1 Signal and image processing

Sparse representations are extensively used in image denoising, compression, and restoration, where signals can be expressed using a small number of basic elements from learned or predefined dictionaries. Low-rank models, such as Principal Component Analysis (PCA) and Robust PCA (RPCA), are commonly employed in background subtraction, face modelling, and video surveillance.

7.2 Audio and speech processing

Parsimonious models have proven highly effective for tasks such as source separation, speech enhancement, and music analysis. Techniques based on sparse coding and Non-negative Matrix Factorization (NMF) allow the decomposition of audio signals into meaningful components, such as vocals and accompaniment. Robust extensions further improve performance in noisy environments by handling outliers and non-stationary interference.

7.3 Pattern recognition and computer vision

Sparse and structured sparse representations enable effective feature extraction for object recognition, activity recognition, and facial analysis. By encoding data using a limited number of informative features, these models enhance discrimination while reducing sensitivity to noise and irrelevant variations.

7.4 domain of biomedical signal analysis

Tasks such as electroencephalogram (EEG) and electrocardiogram (ECG) analysis, medical imaging, and genomics. Low-rank and sparse assumptions help uncover latent physiological patterns, suppress noise, and reduce dimensionality. These properties are particularly valuable in medical applications, where interpretability, robustness, and reliability are critical.

7.5 Anomaly detection and data mining

Low-rank models can characterize normal behaviour in large datasets, while sparse components capture anomalies or rare events. This approach has been successfully applied in network monitoring, fault detection, and financial data analysis.

8. Challenges And Future Scope

8.1 Model–Data Mismatch:

Parsimonious models often rely on ideal assumptions such as exact sparsity or low-rank structure, which real-world data rarely satisfy. This mismatch can lead to performance degradation when signals deviate from the assumed structure.

8.2 Computational Complexity:

Traditional parsimonious methods depend on iterative optimization, resulting in high computational cost, data-dependent runtime, and energy inefficiency, limiting their applicability to real-time and low-power systems.

8.3 Non-Convex Optimization:

Joint learning of representations and dictionaries leads to non-convex problems, causing sensitivity to initialization, convergence to local minima, and lack of strong optimality guarantees.

8.4 Sensitivity to Noise and Misalignment:

Parsimonious models are vulnerable to noise, outliers, and data misalignment. Although robust variants exist, they typically increase computational and model complexity.

8.5 Model Selection and Parameter Tuning:

Choosing appropriate regularization parameters, sparsity levels, and rank bounds is problem-specific and often requires extensive tuning, which affects robustness and usability.

8.6 Limited Expressiveness Compared to Deep Models:

While interpretable, parsimonious models may struggle to capture complex nonlinear structures without hybrid or learning-based extensions.

8.7 Future Research Directions:

Promising avenues include hybrid parsimonious–deep architectures, automated and adaptive regularization, hardware-aware optimization for edge and IoT devices, multimodal parsimonious representations, and stronger theoretical guarantees for learned and online pursuit processes

9. Conclusion

This work has provided a comprehensive exploration of parsimonious models, emphasizing their role as a principled and efficient approach to representing complex, high-dimensional data. By leveraging

sparsity, structured sparsity, and low-rank assumptions, parsimonious models enable compact representations that enhance interpretability, robustness to noise, and generalization performance across a wide range of applications. The study highlighted the limitations of traditional optimization-based parsimonious methods, particularly their high computational cost, unpredictable runtime, and limited suitability for real-time and low-power environments. To address these challenges, learnable pursuit processes were introduced as a process-centric alternative that transforms iterative optimization algorithms into fixed-complexity, trainable architectures. This approach preserves the mathematical structure of classical methods while enabling fast, deterministic inference and efficient deployment on resource-constrained platforms. The integration of supervised learning, discriminative objectives, and geometric transformation optimization further extends the applicability of parsimonious models beyond pure reconstruction tasks. These extensions allow models to adapt to real-world data characteristics, handle misalignment and outliers, and perform task-driven learning for applications such as classification, source separation, and activity recognition. Online learning capabilities additionally ensure adaptability in streaming and large-scale data scenarios.

In conclusion, parsimonious models represent a powerful and versatile framework that bridges the gap between classical signal processing and modern machine learning. By combining model-based rigor with data-driven learning, they offer an effective solution for achieving efficient, interpretable, and scalable representations. As computational demands continue to grow and resource constraints become increasingly important, parsimonious models—particularly those based on learnable pursuit processes—are poised to play a significant role in the future of intelligent, low-power, and real-time systems.

10. References

1. “Edge AI in Practice: A Survey and Deployment Framework,” *Electronics*, 2025 — reviews pruning, quantization, and lightweight architectures tailored for edge AI.
2. X. Wang et al., “A Comprehensive Survey on On-Device AI Models,” *arXiv*, 2025 — analysis of on-device AI including model compression and sparse methods.
3. “Deploying AI on Edge: Advancement and Challenges in Lightweight AI Models,” *Mathematics*, 2025 — highlights enabling techniques like sparsity and quantization for low-power devices.
4. “Quantization and Pruning Strategies for Energy-Efficient TinyML Models,” *ResearchGate*, 2025 — reviews state-of-the-art pruning and quantization for ultra-low-power TinyML.
5. “Model Compression for Deep Neural Networks: A Survey,” *Computers*, 2023 — foundational survey of compression techniques relevant for energy-efficient edge models.
6. S. Francy and R. Singh, “Edge AI: Evaluation of Model Compression Techniques for Convolutional Neural Networks,” *arXiv*, Sep. 2024 — structured pruning and quantization evaluated on edge devices.
7. RAMAN: A Re-configurable and Sparse tinyML Accelerator for Inference on Edge, *arXiv*, 2023 — architecture leveraging sparsity for energy-efficient edge inference.
8. “Lightweight Deep Learning Models for Edge Devices — A Survey,” *ResearchGate*, 2025 — a comprehensive review of compression and optimization for edge AI.
9. S. Kundu et al., “Pre-defined Sparsity for Low-Complexity Convolutional Networks,” *arXiv*, 2020 — sparseness for reduced model complexity.
10. J. Moosmann et al., “Flexible and Fully Quantized Ultra-Lightweight TinyissimoYOLO for Ultra-Low-Power Edge Systems,” *arXiv*, 2023 — ultra-lightweight CNN for low-power systems.
11. Optimization and Security of AI Models for Deployment at Edge Devices, *IEEE* 2025 — discusses



lightweight architectures like MobileNet and TinyML