The Deterministic Seed-Manifold and the Paradigm of Weight-Agnostic Neural Compression

The Paradigm Shift in Neural Representation

Historically, the capacity and computational efficacy of artificial neural networks have been inextricably linked to the continuous optimization of high-dimensional weight matrices. In this standard regime, a vast parameter space is iteratively fine-tuned via stochastic gradient descent and backpropagation. However, an accelerating vector of computational research challenges the fundamental necessity of explicitly storing, transporting, and updating every scalar value within these matrices.

The loom_seed_mnist Proof-of-Concept represents a radical exploration into the absolute limits of procedural weight generation[1]. Rather than treating synaptic weights as free parameters, this architecture enforces a strict mathematical constraint: every dense weight matrix is the deterministic output of a Pseudo-Random Number Generator initialized by a single 64-bit integer, known as a layer seed. Consequently, the entire parameter space of a dense multi-layer perceptron is collapsed into a microscopic manifold defined exclusively by these seed values[1].

This analysis provides a comprehensive, multi-tiered examination of the Loom seed-based experiment, the mechanics of its non-differentiable training algorithms, and its underlying execution engine. Furthermore, the report contextualizes this deterministic approach within the broader academic landscape of weight-agnostic neural networks, the Strong Lottery Ticket Hypothesis, and pseudo-random basis compression methodologies. Ultimately, this synthesis projects how seed-driven architectures and biological target propagation will reshape edge computing, neuroevolution, and hardware-accelerated inference.

Deconstructing the Loom Seed MNIST Proof-of-Concept

The loom_seed_mnist framework is explicitly constructed as an experimental divergence from standard gradient-based training protocols[1]. Its primary objective is not to achieve state-of-the-art accuracy on the MNIST dataset, but rather to empirically test how far a neural network's accuracy can be pushed when the solely mutable parameters are the initial seeds dictating the He-initialization algorithms of each individual layer.

The experimental setup isolates the variables by utilizing a fixed, dense multi-layer perceptron architecture comprising an input layer of 784 nodes, a hidden layer of 128 nodes, a subsequent hidden layer of 64 nodes, and a final linear logit layer of 10 nodes[1]. In a standard 32-bit

floating-point implementation, this topology requires over one hundred thousand distinct parameters. Under the Loom procedural constraint, the network's entire operational state is encapsulated by exactly three unsigned 64-bit integers, representing one seed for each dense layer[1]. Checkpoints are thereby stored as highly compact JSON structures comprising mere hundreds of bytes, entirely bypassing the necessity for dense binary weight blobs[1].

The mathematical mapping defined by this framework is strictly deterministic. A 64-bit seed initializes an xorshift64* pseudo-random number generator, which subsequently draws values from a normal distribution scaled by the He-initialization variance[1]. Because this projection from a single integer to a high-dimensional matrix is a one-way, chaotic function, the resulting loss landscape is rendered entirely non-differentiable with respect to the genomic seeds.

Optimization Regimes in Discrete Seed Space

To navigate this chaotic, non-differentiable discrete search space, the framework deploys three distinct black-box evolutionary search modes[1].

Search Paradigm	Operational Mechanism	Validation Peak	Key Experimental Insight
Warmth	Single-Genome Hill Climb with 64-bit probing and soft fitness evaluation.	~15.0%	Chaotic avalanche effects prevent continuous gradients; local "warm bits" rapidly go cold.
DNA	Clustered Island Model utilizing cosine similarity topological collapse metrics.	~18.8%	Tracking full validation sets and injecting immigrant DNA prevents premature clone death.
DNA-Layer	Coordinate Genetic Ascent focusing population capacity on single layers sequentially.	~18.3%	Feature extraction variance in the largest initial layer dictates the entire performance ceiling.

Warmth: Soft-Fitness Hill Climbing

The warmth mode operates as a highly localized compass on a single, isolated set of seeds. Because hard classification accuracy tends to plateau instantly around chance levels for purely random initializations, the algorithm utilizes a specialized soft fitness metric. This metric combines Softmax Negative Log-Likelihood with a margin penalty to detect microscopic directional shifts in the logit distribution, ensuring that the evolutionary dial continues to move even when hard categorical accuracy is entirely stalled at ten percent[1].

The algorithm iterates layer-by-layer, probing all 64 individual bit flips of a specific integer seed. It measures the delta in soft fitness for each flip to construct a 64-element map of "warm" bits. Mutations are probabilistically biased toward flipping these warm bits, while the algorithm simultaneously evaluates complementary perturbations in integer space via antithetic sampling[1]. Despite the sophisticated local search, the warmth approach proved highly susceptible to local minima. While it successfully elevated the network from a 10.2% baseline to approximately 15% validation accuracy, the warm bits rapidly went cold[1]. The chaotic nature of pseudo-random generators means that a single bit flip does not result in a minor geometric perturbation in the continuous weight space; rather, it results in a completely orthogonal weight matrix. Thus, localized hill-climbing in seed space fundamentally lacks the continuous geographic gradients required for deep convergence.

Clustered Multi-Seed DNA Attract

To overcome the premature convergence inherent in the single-genome approach, the primary DNA mode employs a clustered evolutionary strategy heavily inspired by the NeuroEvolution of Augmenting Topologies and island-model population genetics. The standard configuration initializes four distinct clusters, each containing a local population of six genomes, resulting in twenty-four concurrent models[1]. Each genome represents all three layer seeds moving in tandem.

The evolutionary step updates a candidate seed based on the cluster's elite genome, a crossover mask, and a mutation mask. Crucially, the rates for crossover and mutation are dynamically adjusted based on the topological collapse of the cluster. The framework leverages its internal DNA engine to compute the mean pairwise cosine similarity of the population[1]. If the collapse metric exceeds a threshold of 0.85, the algorithm injects fresh random immigrant DNA into the cluster to force geographic diversity within the search space[1]. If the collapse approaches absolute homogeneity, the crossover rate shrinks to prevent the population from identically cloning the elite genome[1]. This clustered approach proved vastly superior to the localized hill climb, achieving a peak validation accuracy of 18.8% by continuously tracking hall-of-fame updates against the full validation set rather than relying on noisy, batch-level soft fitness[1].

Coordinate DNA-Layer Ascent

The final search mode acts as an architectural ablation study. It applies the clustered DNA mathematics but utilizes a coordinate ascent pattern, freezing the seeds of the subsequent layers and dedicating the entire population's search capacity to optimizing a single layer's seed for a fixed number of generations before rotating to the next[1].

This ablation revealed a critical insight regarding the capacity distribution of the procedurally generated multi-layer perceptron. During multi-round searches, only the primary input layer's seed was ever permanently updated to the hall-of-fame, driving accuracy sequentially from 9.45% up to 18.31%[1]. The mid and head layers remained at their original topology seeds throughout the entire experiment. Because the initial layer contains the overwhelming majority of the parameter volume, finding a statistically advantageous initialization recipe in this specific layer provided enough feature extraction variance that the subsequent, purely random projections could function as adequate linear classifiers[1].

The framework provides comprehensive diagnostic output utilizing deviation metric heatmaps, which generate quality scores ranging from zero to one hundred based on the severity of prediction errors[1]. However, the performance ceiling of approximately 19% highlights the fundamental boundary between discrete seed space and continuous weight space. Searching for favorable initialization seeds is a search for broad statistical recipes rather than the explicit sculpting of parameter weights. The manifold of expressible neural networks under this tight constraint is infinitesimally small compared to an unconstrained Euclidean space, proving that while evolution can identify a seed projecting inputs into a linearly separable subspace, it cannot organically construct the microscopic, fine-tuned feature detectors required for high-fidelity image classification[1].

The Volumetric Infrastructure: M-POLY-VTD Engine

The execution and diagnostic evaluation of the procedural seed experiments rely entirely on the underlying infrastructure of the Loom engine, architected as a Deterministic Neural Virtual Machine[2]. The engine, designated as the Multi-numerical Polymorphic Volumetric Tiled-tensor Dispatcher, provides the exact computational rigidness required to evaluate chaotic integer genomes across heterogeneous hardware topologies[3].

Strict Computational Determinism

A foundational requirement for evaluating pseudo-random weight generations is absolute numerical consistency. The engine guarantees bitwise-identical results across x86 central processing units, ARM processors, WebAssembly environments, and graphics processing units leveraging Apple Metal or WebGPU[2]. By enforcing strict computational determinism, measured by a Mean Absolute Error of less than 10^{-8} across all platforms, the system ensures that a specific genomic seed evaluated on a mobile device yields the exact same fitness score when passed to a distributed donor node over a network[6]. Without this mathematical rigidity, genetic algorithms operating on chaotic functions would immediately fail, as hardware-induced floating-point drift would introduce fatal variance into the fitness evaluations.

Computing Environment	Execution Technology	Output Hash Determinism	Mean Absolute Error vs Reference

Native x86/ARM	Go Binary (Plan 9 SIMD)	Exact Match	0.00000000
Browser Edge	WebAssembly (WASM)	Exact Match	0.00000002
Accelerated Edge	WebGPU / Apple Metal	Exact Match	$< 10^{-8}$
Dedicated Hardware	Intel / Qualcomm NPU	Algorithmic Parity	Vendor Specific Tolerance

Topological Fingerprinting and the DNA Engine

Because the framework treats neural networks as volumetric, three-dimensional grids rather than standard one-dimensional directed acyclic graphs, traditional methods of state comparison fail[3]. The platform integrates a proprietary DNA Engine utilizing principles of topological data analysis[8].

The extraction algorithm converts any layer's weight space into an L2-normalized vector signature[9]. This normalization is highly critical, as it captures the geometric direction of the weights independently of their bit-depth magnitude. Through cosine similarity functions, the engine can accurately compare a 32-bit floating-point layer to a heavily quantized 8-bit integer layer, scoring near-perfect similarity if the underlying topological logic remains preserved[8].

This fingerprinting system extends to identifying logic shifts, instances where an evolutionary algorithm or topological search migrates a functional sub-network to a completely different spatial coordinate within the three-dimensional grid[8]. In the context of population genetics, this metric is exactly what allows the clustered DNA algorithm to monitor spatial collapse and inject immigrant mutations before the search stagnates[1].

Biological Learning and Target Propagation

While the procedural seed framework bypasses weight-tuning entirely by searching discrete integers, the broader execution engine relies on an alternative to stochastic gradient descent known as target propagation[2]. This methodology significantly bridges the gap between artificial optimization and biologically plausible learning.

Standard backpropagation suffers deeply from the weight transport problem, requiring exact mathematical symmetry between forward and backward neural pathways, which necessitates the transmission of signed error signals[11]. This strict requirement is heavily criticized within experimental neuroscience, as biological brains definitively lack symmetric bidirectional synaptic connections[11].

Target propagation resolves this physiological impossibility by propagating idealized target activations rather than explicit mathematical gradients[11]. If the output of a network deviates from the ground-truth label, target propagation generates a target output shifted geometrically in the negative gradient direction[11]. This target is then propagated backward through approximate inverse functions, representing distinct feedback pathways, to generate localized targets for each hidden layer[11]. The forward synaptic weights are subsequently updated locally to map the original activations closer to these explicitly computed targets. Advanced forms of this theory, such as Difference Target Propagation, introduce local difference reconstruction loss to stabilize training across non-invertible feedforward functions[15]. Mathematically, target propagation has been demonstrated to be closely aligned with Gauss-Newton optimization rather than standard gradient descent[11]. This makes it exceptionally effective for discrete-time neural meshes where standard autograd graphs fail to traverse spatial feedback loops efficiently[11]. The engine's implementation of this protocol allows layers to learn via asynchronous, localized Hebbian updates in real-time, completely bypassing the requirement to freeze the entire network state during a backward pass[8].

Weight-Agnostic Topologies and Network Pruning

The procedural generation of weights via specific seeds represents only one intersection of a much broader academic pursuit focused on reducing or entirely eliminating the continuous parameter space of artificial neural networks. Analyzing alternative weight-agnostic methodologies provides necessary context for where deterministic seed generation currently resides within machine learning literature.

Weight Agnostic Neural Networks

Where the procedural seed framework fixes the network topology and evolves the initialization algorithm, Weight Agnostic Neural Networks invert the paradigm by fixing the weights and evolving the architectural topology itself[16].

Research into weight-agnostic structures challenges the deeply held assumption that architectural inductive biases are inherently secondary to continuously learned weights. Utilizing evolutionary algorithms designed to augment topologies, researchers begin with minimal, sparsely connected architectures. During the evaluation phase, instead of training the synaptic connections, a single shared weight value is applied universally to every connection within the network[16].

Remarkably, these topological networks have demonstrated the capability to solve continuous control reinforcement learning tasks entirely devoid of gradient updates[16]. On supervised learning tasks such as the MNIST dataset, weight-agnostic topologies have achieved accuracy exceeding ninety percent when an ensemble of networks, each assigned a different shared uniform weight, is deployed[4].

The success of these networks relies heavily on the Baldwin Effect, a concept from evolutionary biology that describes the selective pressure rewarding structures that are predisposed to successful behaviors without requiring computationally expensive learning phases[4]. While the

dense seed initialization experiment struggled to surpass a nineteen percent accuracy threshold, weight-agnostic topologies prove that if the network architecture is allowed to become highly irregular and uniquely specialized to the dataset, explicit fine-tuning of scalar weights becomes mathematically unnecessary. This evidence suggests that the optimal future application for procedural seed engines lies not in initializing massive dense layers, but in coupling integer seed-generation with aggressive topological mutation algorithms[7].

Paradigm	Search Target	Fixed Element	Evolutionary Mechanism	Inference Reconstruction
Procedural Seeds	64-bit Integers	Topology & Dense Mapping	Island Model Cosine Tuning	He_Init(Optimal Seed)
Weight Agnostic	Network Topology	Shared Uniform Weight	Topology Augmentation	Evolved Topology + Shared Scalar
Strong Lottery	Binary / Ternary Mask	Random Base Initialization	Pruning and Sign Flipping	Learned Mask * PRNG(Seed)

The Strong Lottery Ticket Hypothesis and Supermasks

The original Lottery Ticket Hypothesis posits that dense, randomly initialized neural networks contain highly sparse subnetworks that can match the precise performance of the full model when trained in isolation[19]. The Strong Lottery Ticket Hypothesis extends this concept to its absolute theoretical extreme: these sparse subnetworks exist in a state that can perform at high accuracy without undergoing any continuous weight training whatsoever[21].

Instead of iteratively updating continuous weights, algorithms leveraging this hypothesis learn a binary supermask, consisting solely of zeros and ones, which is permanently superimposed over a frozen, pseudo-randomly initialized matrix[19]. A highly advanced variant known as the Trichromatic Strong Lottery Ticket Hypothesis expands this singular mask into three distinct primary supermasks to control structural connectivity, sign flipping, and magnitude scaling independently[21].

By exploring the parameter space through masking rather than mathematical adjustment, trichromatic supermasks have achieved profound compression ratios. For instance, a ResNet-50 model trained on the CIFAR-100 dataset was compressed by a factor of thirty-eight down to 2.51 megabytes, while still maintaining an impressive 78.43% accuracy[21].

These supermasks confirm that the initial state of a pseudo-random number generator contains a vast, highly capable distribution of features, provided the optimization algorithm

possesses the capability to select the optimal subset of that distribution. The procedural seed experiment restricted itself by evaluating the entirety of a pseudo-random output via dense layers, severely limiting its accuracy[1]. If the seed architecture were analytically augmented with a trainable binary mask, effectively merging the Strong Lottery Ticket Hypothesis with deterministic procedural generation, the performance ceiling on supervised tasks would likely shatter current limitations[1].

Pseudo-Random Basis Compression: PRANC, NOLA, and SeedLM

The most mature and commercially viable implementations of seed-driven architectures currently manifest in advanced compression techniques, specifically the Pseudo Random Networks for Compacting deep models (PRANC), Low-Rank Adaptation decoupling (NOLA), and SeedLM[24].

Instead of treating a single deterministic seed as the sole defining characteristic of an entire dense layer, these frameworks represent a highly complex deep model as a linear combination of several randomly initialized, permanently frozen basis networks[27].

In this mathematical formulation, a target weight block is approximated as the sum of multiple pseudo-random matrices, each generated efficiently on-the-fly via Linear Feedback Shift Registers initialized by a specific, carefully selected scalar seed[24]. During the training or post-training compression phase, the algorithm optimizes only the continuous linear mixture coefficients associated with each basis matrix, while simultaneously searching for the optimal generative seed[24].

This hybrid approach elegantly bridges the stark divide between discrete seed space and continuous gradient optimization. SeedLM has proven capable of compressing massive architectural weights, such as the Llama 3 70-Billion parameter model, down to three-bit and four-bit equivalents without requiring any external calibration data[24]. Astoundingly, this approach retains approximately 97.9% of the original model's zero-shot accuracy across diverse tasks, outperforming state-of-the-art quantization techniques[24].

Similarly, the PRANC framework has demonstrated the ability to condense standard image classification models by a staggering factor of one hundred while maintaining acceptable task performance[25]. For example, an AlexNet model comprising 1.7 million parameters can be compacted down to a mere ten thousand parameters via pseudo-random basis generation, far exceeding the efficiency of traditional dataset distillation methods[28].

Furthermore, the NOLA framework adapts this pseudo-random foundation to revolutionize Low-Rank Adaptation. Standard adaptation methods are mathematically bounded by a rank-one matrix decomposition, strictly limiting the extent of compression based on the model's inherent architecture[26]. NOLA systematically bypasses this lower bound by reparameterizing the low-rank matrices as linear combinations of randomly generated basis matrices[26]. By optimizing only the mixture coefficients, NOLA entirely decouples the trainable parameter count from the network architecture. Experimental data reveals that NOLA can

achieve comparable performance to standard Low-Rank Adaptation while utilizing twenty times fewer parameters, successfully fine-tuning a seventy-billion parameter language model with a budget of just 0.6 million parameters[32].

Extreme Evolutionary Strategies in High-Dimensional Search

While backpropagation via stochastic gradient descent remains the absolute industry standard, the rigorous exploration of evolutionary strategies for neural network optimization demonstrates that non-differentiable, random-search approaches can scale effectively under specific constraints.

Academic investigations into hyperparameter tuning frequently reveal that purely random searches often outperform exhaustive grid searches. Because high-dimensional objective functions typically exhibit a low effective dimensionality—meaning the output is far more sensitive to changes in a few specific dimensions than others—randomly sampled trials provide a significantly more efficient coverage of the critical subspaces[36]. This mathematical reality underpins why genetic algorithms consistently excel at discovering optimal network configurations[36].

In specialized medical diagnostic fields, combinations of covariance matrix adaptation evolution strategies have achieved profound results. A weight-agnostic neural network applied to breast cancer risk assessment data achieved state-of-the-art classification performance, yielding an F1-score of 0.933 while utilizing a topology restricted to a mere 163 synaptic connections[38].

Moreover, pure genetic selection mechanisms lacking any gradient descent have demonstrated the capability to achieve up to 81.47% accuracy on the MNIST dataset, operating in strictly unsupervised or continuous reinforcement contexts[39]. These evolutionary models reveal that genetic pressure forces networks into minimal, highly strategic parameter utilization. For example, a heavily constrained 32-neuron genetic model will automatically prioritize the absolute mastery of simpler visual digits over complex, highly variable digits due to strict capacity limits[39]. This presents an optimization landscape that is distinctly and fundamentally different from the generalized, uniform minima sought by traditional stochastic gradient descent[39].

The integration of these evolutionary dynamics into spiking neural networks further enhances their viability. Liquid State Machines evolved to exhibit biologically plausible small-world topological properties—characterized by densely connected local hub nodes and long-tail degree distributions—have achieved classification accuracies of 98.12% on image recognition tasks, drastically improving energy efficiency and computational streamlining without requiring symmetric error propagation[40].

Strategic Implications and Future Operational Trajectories

The successful fusion of deterministic neural virtual machines, pseudo-random basis generation, and biologically plausible target propagation points directly toward several profound paradigm shifts in the deployment and architecture of artificial intelligence.

Eradicating the Memory Bandwidth Wall at the Edge

The primary, insurmountable bottleneck for executing massive language models and volumetric neural networks on consumer hardware is not the availability of arithmetic logic units, but rather fundamental memory bandwidth[5]. Fetching gigabytes of floating-point weights from video random-access memory to the processor stalls inference engines on nearly all edge devices.

Seed-based architectures conceptually invert this physical hardware constraint[25]. By utilizing deterministic algorithms, such as linear feedback shift registers, to generate dense weight blocks dynamically on-the-fly, these frameworks strategically trade inherently idle computational cycles for a massive reduction in continuous memory reads[24]. A model reconstructed from random seeds and highly sparse linear coefficients can theoretically execute entirely within the localized L1 and L2 caches of modern architectures. The required matrices are generated exactly at the moment of the necessary matrix multiplication operation and immediately discarded[24]. As global model sizes inexorably push toward the trillion-parameter threshold, on-the-fly programmatic weight instantiation will become a critical, unavoidable path for mobile, embedded, and edge deployment.

Cryptographic Model Federation and Zero-Knowledge Transfer

Traditional federated learning systems require the constant transmission of continuous weight deltas across open networks, leaving them highly susceptible to interception, data poisoning, or reverse-engineering attacks. Conversely, seed-driven architectures inherently possess robust cryptographic properties[25].

Because pseudo-random number generators, specifically shift registers and cryptographic hash functions, exhibit a severe avalanche effect, any generated sequence maintains absolutely minimal statistical correlation with even a microscopically shifted version of itself[28]. An artificial intelligence model defined by a complex mixture of pseudo-random basis networks can therefore be securely transmitted across public networks by sharing only the continuous linear coefficients; the critical 64-bit generative seeds act as entirely private decryption keys[25]. If an unauthorized party attempts to reconstruct the neural model with an incorrect seed—even one derived a single integer away from the true value—the generated basis networks will be completely orthogonal, rendering the intercepted coefficients useless and ensuring the model's output remains completely indistinguishable from stochastic noise[8].

The Convergence of Biological Plausibility and Continuous Edge Learning

The combination of target propagation and evolutionary topological shifts offers a highly biologically plausible and computationally viable model for continuous, lifelong edge learning[10].

Unlike backpropagation, which demands that the entire global network state be mathematically frozen during a backward pass to compute gradients, target propagation allows highly localized neural modules to learn asynchronously based purely on target activations[11].
When combined with discrete-time architectural systems featuring double-buffering and asynchronous layer propagation, these networks can learn via localized Hebbian updates in real-time, matching the operational realities of biological organisms[8]. Furthermore, topological fingerprinting allows a decentralized system to track precisely how functional logic migrates across a three-dimensional neural grid in response to continuous data stimuli, effectively replicating advanced neuroplasticity without the crushing computational overhead of traditional deep learning frameworks[3].
By completely replacing gigabytes of static memory architecture with deterministic, algorithmic generation, and replacing fragile backpropagation with robust target propagation, the industry is establishing a clear trajectory toward highly secure, infinitely portable, and ultra-compressed artificial intelligence capable of residing autonomously on the absolute periphery of the global compute ecosystem.

Works cited

1. loom_seed, uploaded:loom_seed
2. Loom - Layered Omni-architecture Openfluke Machine - GitHub, https://github.com/openfluke/loom
3. M-POLY-VTD: Architecture Overview — Loom Docs - OpenFluke, https://openfluke.com/docs/overview
4. Exploring Weight Agnostic Neural Networks - Google Research, https://research.google/blog/exploring-weight-agnostic-neural-networks/
5. loom/README.md at main · openfluke/loom - GitHub, https://github.com/openfluke/loom/blob/main/README.md
6. LOOM: The Universal AI Runtime That Works Everywhere (And Why That Matters) - Medium, https://medium.com/@planetbridging/loom-the-universal-ai-runtime-that-works-everywhere-and-why-that-matters-54de5e7ec182
7. loom/docs/index.md at main · openfluke/loom - GitHub, https://github.com/openfluke/loom/blob/main/docs/index.md
8. Loom M-POLY-VTD — Golang AI Architecture Deep Research - OpenFluke, https://openfluke.com/loom/research
9. Quick Reference: Common Code Snippets — Loom Docs | OpenFluke, https://openfluke.com/docs/quick-reference
10. Why Loom? v0.83 SIMD, Metal, OpenVINO & Edge AI - OpenFluke, https://openfluke.com/why-loom
11. A Theoretical Framework for Target Propagation, https://proceedings.neurips.cc/paper/2020/file/e7a425c6ece20cbc9056f98699b53c6f-Paper.pdf
12. Backpropagation and the brain - Oxford University Research Archive,

https://ora.ox.ac.uk/objects/uuid:862189c1-0088-4f78-b17a-2748c2019209/download_file?safe_filename=Lillicrap_v6_2020.pdf
13. Target Propagation via Regularized Inversion - arXiv, https://arxiv.org/pdf/2112.01453
14. (PDF) A Theoretical Framework for Target Propagation - ResearchGate, https://www.researchgate.net/publication/345723487_A_Theoretical_Framework_for_Target_Propagation
15. Efficient Target Propagation by Deriving Analytical Solution, https://ojs.aaai.org/index.php/AAAI/article/view/28977/29858
16. Weight Agnostic Neural Networks - GitHub Pages, https://spartee.github.io/NeurIPS-2019/weight-agnostic-nn.pdf
17. (PDF) Weight Agnostic Neural Networks - ResearchGate, https://www.researchgate.net/publication/333717007_Weight_Agnostic_Neural_Networks
18. Weight Agnostic Neural Networks, https://weightagnostic.github.io/
19. Pruning Neural Networks with Supermasks, https://www.esann.org/sites/default/files/proceedings/2021/ES2021-126.pdf
20. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask - Uber, https://www.uber.com/us/en/blog/deconstructing-lottery-tickets/
21. The Trichromatic Lottery Ticket Hypothesis | NTT Communication Science Laboratories, https://www.rd.ntt/e/cs/team_project/media/recognition/research_media23.html
22. Multicoated Supermasks Enhance Hidden Networks - Proceedings of Machine Learning Research, https://proceedings.mlr.press/v162/okoshi22a/okoshi22a.pdf
23. The Trichromatic Strong Lottery Ticket Hypothesis: Neural Compression With Three Primary Supermasks - NeurIPS 2026, https://neurips.cc/virtual/2024/98237
24. SeedLM: Compressing LLM Weights into Seeds of Pseudo-Random Generators - arXiv, https://arxiv.org/html/2410.10714v1
25. PRANC: Pseudo RAndom Networks for Compacting deep models, https://web.cs.ucdavis.edu/~hpirsiav/papers/pranc_iccv23.pdf
26. NOLA: Compressing LoRA using Linear Combination of Random Basis, https://ucdvision.github.io/NOLA/
27. PRANC: Pseudo RAndom Networks for Compacting deep models - CVF Open Access, https://openaccess.thecvf.com/content/ICCV2023/papers/Nooralinejad_PRANC_Pseudo_RAndom_Networks_for_Compacting_Deep_Models_ICCV_2023_paper.pdf
28. PRANC: Pseudo RAndom Networks for Compacting deep models - OpenReview, https://openreview.net/pdf?id=YvBCq9gAeX
29. [2206.08464] PRANC: Pseudo RAndom Networks for Compacting deep models - arXiv, https://arxiv.org/abs/2206.08464
30. SEEDLM: COMPRESSING LLM WEIGHTS INTO SEEDS OF PSEUDO-RANDOM GENERATORS - ICLR Proceedings, https://proceedings.iclr.cc/paper_files/paper/2025/file/222a2a46018a1e7b55ba48ba11932d04-Paper-Conference.pdf

31. SeedLM: Compressing LLM Weights into Seeds of Pseudo-Random Generators - arXiv, https://arxiv.org/pdf/2410.10714
32. NOLA: Compressing LoRA using Linear Combination of Random Basis - Liner, https://liner.com/review/nola-compressing-lora-using-linear-combination-of-random-basis
33. nola: compressing lora using linear combination of random basis - arXiv, https://arxiv.org/pdf/2310.02556
34. NOLA: Compressing LoRA using Linear Combination of Random Basis - GitHub, https://github.com/UCDvision/NOLA
35. [2310.02556] NOLA: Compressing LoRA using Linear Combination of Random Basis - arXiv, https://arxiv.org/abs/2310.02556
36. Random Search for Hyper-Parameter Optimization - Journal of Machine Learning Research, https://jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf
37. Ensemble genetic and CNN model-based image classification by enhancing hyperparameter tuning - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC11704345/
38. Dynamic Weight Agnostic Neural Networks and Medical Microwave Radiometry (MWR) for Breast Cancer Diagnostics - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC9497764/
39. [R]Evolution vs Backprop: Training neural networks through genetic selection achieves 81% on MNIST. No GPU required for inference. - Reddit, https://www.reddit.com/r/deeplearning/comments/1pz0hit/revolution_vs_backprop_training_neural_networks/
40. Emergence of brain-inspired small-world spiking neural network through neuroevolution - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC10847652/