



Advanced Methods in Tensor Network Decomposition and Structure Learning

Master's Thesis

Dipartimento di Automatica e Informatica (DAUIN)
Artificial Intelligence and Data Analytics

Abolfazl Javidian (s314441)

February 2026

Supervisors: Prof. Gianpiero Cabodi, Prof. Rahman Tashakkori

Acknowledgements

I would like to express my sincere gratitude to my supervisors for their guidance, encouragement, and continuous support throughout the development of this thesis.

First and foremost, I would like to thank my first supervisor, **Professor Gianpiero Cabodi** from the *Politecnico di Torino, Italy*, for his valuable advice, insightful discussions, and for providing me the opportunity to work on this research topic. His expertise and constructive feedback have played a crucial role in shaping this work.

I would also like to extend my sincere thanks to my second supervisor, **Professor Rahman Tashakkori** from *Appalachian State University, Boone, North Carolina, USA*, for his guidance, support, and thoughtful suggestions during the course of this research.

I am deeply grateful to my brother, **Mohammad Ali Javidian**, whose encouragement, understanding, and constant support have always motivated me during this journey.

Finally, I would like to thank all the people who supported and encouraged me throughout my academic path.

Dedication

This thesis is lovingly dedicated to the memory of my beloved father, whose passing occurred shortly after my migration to Italy.

Although he is no longer with us, his guidance, values, and the lessons he instilled in me continue to shape my life and inspire my efforts. His belief in education and perseverance has always been a source of strength for me. I hope this work makes his spirit proud.

I also dedicate this work to my dear mother, whose unconditional love, patience, and endless support have been the foundation of everything I have achieved. Her sacrifices, encouragement, and faith in me have made this journey possible.

Without her strength and support, this accomplishment would not have been possible.

Abstract

Tensor networks (TNs) provide structured low-rank representations for high-dimensional data and have become important tools in scientific computing and machine learning. This thesis investigates adaptive tensor network topology learning, focusing on topology as an explicit inductive bias that governs both expressivity and computational cost. While classical formats such as Tensor-Train, Tensor-Ring, and Hierarchical Tucker are efficient, their fixed structures restrict flexibility. To address this limitation, the present work develops a framework for feasibility-aware topology adaptation under explicit computational constraints.

A local topology search strategy is proposed for arbitrary-graph tensor networks. Network structure is encoded as a binary edge representation and explored through neighborhood-based modifications. Candidate topologies are filtered using structural feasibility constraints before evaluation. To balance competing objectives, a robust min-max scalarization is introduced, jointly optimizing reconstruction error, compression ratio, and perceptual quality metrics such as PSNR and SSIM. Training combines Adam optimization with optional L-BFGS refinement and gradient clipping to ensure stability.

Experiments on image reconstruction tasks demonstrate that adaptive topologies can outperform canonical chain-based networks while remaining computationally tractable. The results confirm that topology acts as a controllable structural prior affecting both representational efficiency and contraction complexity. The thesis concludes by discussing limitations in scalability and training cost, and outlines future directions including Bayesian-guided structure search and stronger integration with probabilistic modeling frameworks.

Keywords: Tensor Networks; Topology Search; Structure Learning; Min-Max Optimization; Image Reconstruction; Compression; Inductive Bias.

Contents

1	Introduction	1
1.1	Tensors, Tensor Decompositions, and Why Compression Matters . . .	1
1.1.1	Tensors as Multidimensional Data Structures	1
1.1.2	The Curse of Dimensionality	2
1.1.3	Low-Rank Structure in High-Dimensional Problems	3
1.1.4	Classical and Modern Tensor Decomposition Formats	3
1.1.5	Why Compression Matters	5
1.2	What Is a Tensor Network? Origins in Many-Body Physics	6
1.2.1	Historical Origins: DMRG and Quantum Many-Body Systems	6
1.2.2	Matrix Product States and Tensor Network Structure	6
1.2.3	General Tensor Networks and Higher-Dimensional Extensions	7
1.2.4	Modern Viewpoint: Tensor Networks as Computational Models	7
1.3	Canonical Tensor Network Topologies and Their Restrictions	8
1.3.1	Chain Tensor Networks: Matrix Product States and Tensor Trains	8
1.3.2	Tree Tensor Networks: TTN and Hierarchical Tucker	9
1.3.3	Two-Dimensional and Lattice Tensor Networks: PEPS	9
1.3.4	Multi-Scale Tensor Networks: MERA	11
1.3.5	Cyclic Tensor Networks: Tensor Ring	11
1.3.6	Summary of Topological Trade-Offs	12
1.4	Where Tensor Networks Sit in Modern AI and Machine Learning . . .	13

1.4.1	Tensor Networks as Efficient Models	13
1.4.2	Tensor Networks as Probabilistic Representations	13
1.4.3	The Role of Topology and Inductive Bias	14
1.5	Thesis Problem Statement and Contributions	14
1.5.1	Thesis Contributions	15
1.5.2	Thesis Organization	16
2	Related Work	17
2.1	Formulating Tensor Network Topology Search	17
2.1.1	Topology as a Discrete Structural Variable	17
2.1.2	Topology and Contraction Complexity	18
2.1.3	Expensive Evaluation and Search Challenges	18
2.2	Evolutionary and Heuristic Topology Search	19
2.2.1	Evolutionary Formulation	19
2.2.2	Heuristic Search Characteristics	19
2.2.3	Limitations and Open Challenges	20
2.2.4	Positioning Within the State of the Art	20
2.3	Local and Permutation-Based Structure Search	21
2.3.1	Permutation Search as a Structural Optimization Problem	21
2.3.2	Efficiency and Practical Advantages	21
2.3.3	Limitations of Permutation-Based Search	22
2.3.4	Positioning Within the State of the Art	22
2.4	Alternating and Neighborhood Exploration Methods	22
2.4.1	Tensor Network Architecture Learning (TnALE)	23
2.4.2	Advantages of Alternating Structure–Parameter Optimization	23
2.4.3	Limitations and Open Issues	23
2.4.4	Positioning Within the State of the Art	24
2.5	Bayesian Structure Search and Uncertainty-Aware Selection	24
2.5.1	Bayesian Tensor Network Structure Search	24

2.5.2	Surrogate-Based Search with Gaussian Processes	25
2.5.3	Advantages of Uncertainty-Aware Structure Search	25
2.5.4	Limitations and Open Challenges	25
2.5.5	Positioning Within the State of the Art	26
2.6	Adaptive and Greedy Tensor Network Structure Learning	26
2.6.1	Greedy Structure Construction	26
2.6.2	Strengths of the Greedy Approach	27
2.6.3	Limitations and Relation to Other Methods	27
2.6.4	Positioning Within the State of the Art	28
2.7	Limitations of Prior Work and Scope of This Thesis	28
2.7.1	High Evaluation Cost of Structure Search	28
2.7.2	Late Enforcement of Feasibility Constraints	29
2.7.3	Lack of Explicit Multiobjective Optimization	29
2.7.4	Motivation for Min–Max Scalarization	29
2.7.5	Positioning of This Thesis	30
3	Proposed methodology	31
3.1	Problem Setup: Image Tensorization and Reconstruction Objective .	31
3.1.1	Image Tensorization	31
3.1.2	Tensor Network Reconstruction Model	32
3.1.3	Design Implications	32
3.2	Model: Arbitrary-Graph Tensor Network Parameterization	33
3.2.1	Graph-Based Parameterization	33
3.2.2	Binary Encoding of Topology	33
3.2.3	Einsum-Based Contraction Representation	34
3.2.4	Expressivity and Computational Trade-Offs	34
3.2.5	Design Implications	35
3.3	Evaluation Metrics: Reconstruction, Compression, and Perceptual Quality	35

3.3.1	Relative Squared Error (RSE)	35
3.3.2	Compression Ratio	36
3.3.3	Peak Signal-to-Noise Ratio (PSNR)	36
3.3.4	Structural Similarity Index (SSIM)	36
3.3.5	Multiobjective Perspective	37
3.4	Min–Max Objective as Robust Scalarization	38
3.4.1	Multiobjective Optimization Perspective	38
3.4.2	Reference-Normalized Deviations	38
3.4.3	Min–Max Scalarization	39
3.4.4	Feasibility-Aware Evaluation	39
3.4.5	Interpretation and Design Rationale	39
3.5	Topology Search Algorithm	40
3.5.1	Search Space and Neighborhood Definition	40
3.5.2	Feasibility Screening (Safety Constraints)	40
3.5.3	Candidate Evaluation: Train–Reconstruct–Score	41
3.5.4	How the Min–Max Objective Operates During Search	41
3.5.5	Pseudo-code	41
3.5.6	Remarks on Practical Configuration	43
3.6	Topology Search via Feasibility-First Local Exploration	44
3.6.1	Design Principles	44
3.6.2	Local Neighborhood Operator	45
3.6.3	Search Procedure and Acceptance Criterion	45
3.6.4	Relation to Existing Structure Search Methods	45
3.7	Training Routine: Adam to L-BFGS with Gradient Clipping	46
3.7.1	Stage I: Adaptive Optimization with Adam	46
3.7.2	Stage II: Optional Refinement with L-BFGS	46
3.7.3	Gradient Clipping for Stability	47
3.7.4	Practical Considerations	47

4	Experimental results	48
4.1	Experimental Setup	48
4.2	Results for Rank 4	49
4.3	Results for Rank 6	52
4.4	Results for Rank 8	54
4.5	Topology Adaptation in Tensor Networks	57
4.6	Overall Discussion	58
5	Conclusion	60
5.1	What Was Learned About Topology as Inductive Bias	60
5.1.1	Topology Governs Expressivity	60
5.1.2	Topology Determines Computational Cost	61
5.1.3	Balanced Structure Through Robust Optimization	61
5.1.4	Implications for Tensor Network Design	61
5.2	Limitations of the Current Approach	62
5.2.1	Scaling Beyond Eight Nodes	62
5.2.2	Expensive Per-Topology Training	62
5.2.3	Dependence on Thresholds and Weights	63
5.2.4	Contraction Order Not Explicitly Optimized	63
5.2.5	Local Optimality and Limited Exploration Depth	63
5.2.6	Summary of Limitations	64
5.3	Future Directions	64
5.3.1	Bayesian and Surrogate-Guided Structure Search	64
5.3.2	Stronger Integration with Probabilistic Graphical Models	65
5.3.3	Beyond Reconstruction: Generative and Sequence Modeling	65
5.3.4	Scaling and Hierarchical Structure Learning	65
5.3.5	Tensor Networks in Modern AI Systems	66
5.3.6	Summary	66

Chapter 1

Introduction

1.1 Tensors, Tensor Decompositions, and Why Compression Matters

1.1.1 Tensors as Multidimensional Data Structures

A tensor is a multidimensional generalization of scalars, vectors, and matrices [1, 2]. Formally, an N -th order (or N -way) tensor is an element of a tensor product space

$$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N},$$

where I_n denotes the dimension of the n -th mode. Scalars ($N = 0$), vectors ($N = 1$), and matrices ($N = 2$) are special cases of this definition.

Tensors arise naturally in numerous scientific and engineering applications, including signal processing, computational physics, uncertainty quantification, machine learning, and data-driven modeling of complex systems [1]. In many of these contexts, tensor entries represent samples of multivariate functions, discretized solutions of partial differential equations, or high-order correlation data.

While tensors provide a natural and expressive framework for representing multidimensional data, their direct manipulation quickly becomes infeasible as the order N increases due to exponential growth in storage and computational complexity [2].

As shown in Fig. 1.1, tensors generalize lower-order data structures. Figure 1.1a illustrates a scalar, Fig. 1.1b represents a matrix as a second-order tensor, while Fig. 1.1c shows a higher-order tensor with more than two modes.

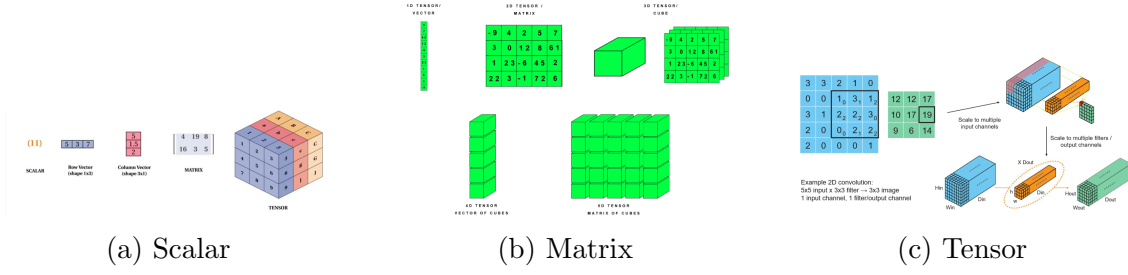


Figure 1.1: Progression from low-order to higher-order tensor objects.

1.1.2 The Curse of Dimensionality

The primary computational challenge associated with tensors is the *curse of dimensionality* [2]. The total number of entries in a dense tensor scales as

$$\prod_{n=1}^N I_n,$$

which grows exponentially with the tensor order. Even for modest mode sizes, storage and computational costs rapidly exceed practical limits.

For example, a tensor of order $N = 10$ with uniform dimension $I_n = 20$ contains $20^{10} \approx 10^{13}$ entries, making explicit storage impossible. This exponential growth affects not only memory requirements but also the complexity of basic operations such as tensor contractions, factorizations, and linear solves.

As a result, naive tensor representations are unsuitable for high-dimensional problems, necessitating alternative representations that exploit hidden structure and separability properties [2].

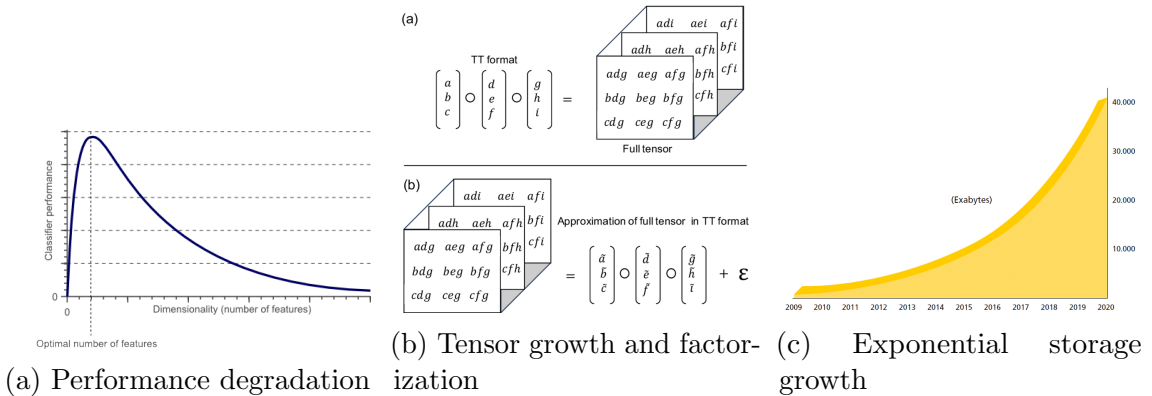


Figure 1.2: Illustration of different manifestations of the curse of dimensionality.

The curse of dimensionality manifests in several complementary ways, as illustrated in Fig. 1.2. Figure 1.2a shows how increasing dimensionality can degrade model

performance, indicating that higher-dimensional representations do not necessarily improve predictive accuracy. Figure 1.2b highlights the rapid growth of a full tensor representation compared to its compressed tensor-train approximation, emphasizing the inefficiency of dense storage. Finally, Fig. 1.2c illustrates the exponential increase in data volume with dimensionality, demonstrating the infeasibility of explicit representations for large-scale problems.

1.1.3 Low-Rank Structure in High-Dimensional Problems

In many practical applications, tensors exhibit significant *redundancy*. Despite their high ambient dimension, the effective degrees of freedom are often much smaller [1]. This observation motivates the concept of *low-rank tensor approximation*.

Low-rank tensor methods seek to approximate a full tensor $\mathcal{X}$ by a structured representation

$$\mathcal{X} \approx \hat{\mathcal{X}},$$

where $\hat{\mathcal{X}}$ depends on significantly fewer parameters. Such approximations exploit correlations across tensor modes and reduce both storage and computational complexity [1, 3].

The notion of rank for tensors is more intricate than in the matrix case. Different tensor formats correspond to different definitions of rank and different trade-offs between expressiveness, numerical stability, and computational efficiency [2].

1.1.4 Classical and Modern Tensor Decomposition Formats

Several tensor decomposition formats have been proposed to address the challenges of high-dimensional representations [1].

Canonical Polyadic (CP) Decomposition. The CP decomposition represents a tensor as a sum of rank-one tensors:

$$\mathcal{X} \approx \sum_{r=1}^R \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)},$$

where $\circ$ denotes the outer product. While conceptually simple, CP decompositions may suffer from ill-posedness and numerical instability [1].

Tucker Decomposition. The Tucker format expresses a tensor via a core tensor multiplied by factor matrices along each mode. Although more stable than CP, the Tucker core grows exponentially with dimension, limiting its applicability to moderate-order tensors [1].

Hierarchical Tucker (HT) Decomposition. The Hierarchical Tucker format introduces a tree-based factorization of tensor modes, reducing storage complexity from exponential to polynomial in N and providing stable approximation properties [3, 2].

Tensor-Train (TT) Decomposition. The Tensor-Train format represents a tensor as a sequence of three-dimensional cores:

$$\mathcal{X}(i_1, \dots, i_N) = G_1(i_1)G_2(i_2) \cdots G_N(i_N),$$

where matrix products replace the full tensor storage. The TT format combines favorable computational properties with strong theoretical guarantees, making it particularly attractive for large-scale numerical problems [4].

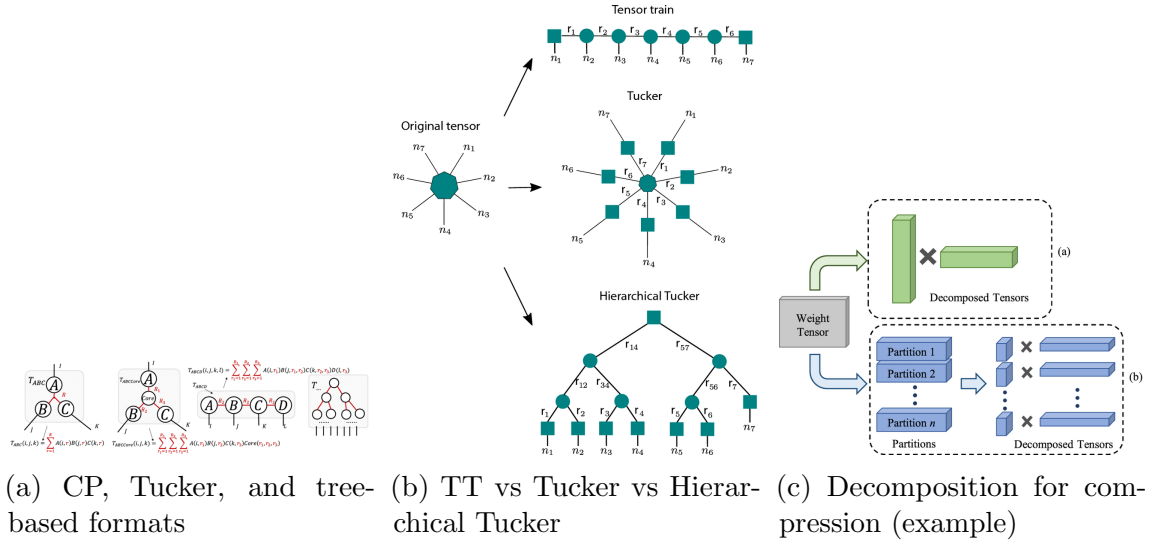


Figure 1.3: Schematic comparison of common tensor decomposition formats and their role in compression.

Figure 1.3 summarizes the structural differences among widely used tensor decompositions. Figure 1.3a contrasts core low-rank models such as CP and Tucker with tree-structured factorizations, highlighting how different rank parameters control the approximation complexity. Figure 1.3b provides an intuitive comparison between

Tensor-Train (TT), which factorizes an N -way tensor into a chain of low-dimensional cores, Tucker, which concentrates interaction information in a central core tensor, and Hierarchical Tucker (HT), which organizes the representation through a dimension tree to avoid exponential core growth. Finally, Fig. 1.3c illustrates how tensor decompositions can be used as compression operators (e.g., for large parameter tensors), either directly or after partitioning, thereby reducing storage and computational cost while maintaining an accurate approximation.

1.1.5 Why Compression Matters

Tensor compression is not merely a matter of reducing memory usage; it fundamentally enables computation in high-dimensional settings [2]. Compressed tensor formats allow:

- Efficient storage of otherwise intractable objects,
- Scalable algorithms for linear algebra and optimization,
- Feasible numerical solutions of high-dimensional PDEs,
- Reduced complexity in machine learning and data analysis.

Moreover, many tensor formats admit quasi-optimal approximations and stable truncation procedures, ensuring controlled approximation errors [3, 4]. This makes tensor decompositions a central tool in modern numerical analysis and scientific computing.

Therefore, tensor decompositions transform the curse of dimensionality into a manageable computational framework by exploiting low-rank structure. Understanding these representations is essential for developing scalable algorithms in high-dimensional problems [1, 2].

1.2 What Is a Tensor Network? Origins in Many-Body Physics

1.2.1 Historical Origins: DMRG and Quantum Many-Body Systems

Tensor networks originated in the study of quantum many-body systems, where the exponential growth of the Hilbert space dimension poses severe computational challenges. A quantum state of N interacting particles is described by a high-order tensor whose size scales exponentially with N , making naive representations intractable.

A major breakthrough was achieved with the introduction of the Density Matrix Renormalization Group (DMRG) by White [5]. DMRG provided an efficient variational algorithm for computing ground states of one-dimensional quantum lattice systems by systematically truncating irrelevant degrees of freedom. Although initially formulated in algorithmic terms, it was later understood that DMRG implicitly constructs a low-rank tensor representation of the quantum state.

This insight led to the identification of the underlying representation as a *matrix product state* (MPS), which provides an explicit tensor network structure capturing the dominant correlations in low-dimensional quantum systems.

1.2.2 Matrix Product States and Tensor Network Structure

Matrix product states express a high-order tensor as a sequence of contracted low-order tensors, forming a one-dimensional network. This representation makes explicit the entanglement structure of quantum states and explains the remarkable success of DMRG in one-dimensional systems.

Schollwöck [6] provided a comprehensive reformulation of DMRG in the language of MPS, establishing a rigorous connection between numerical efficiency and entanglement properties. In particular, MPS efficiently approximate states obeying an *area law* for entanglement entropy, which is typical for ground states of gapped one-dimensional Hamiltonians.

From this perspective, a tensor network can be understood as a graph whose nodes represent tensors and whose edges represent contracted indices. The topology of the graph encodes how correlations and entanglement are distributed across the system.

1.2.3 General Tensor Networks and Higher-Dimensional Extensions

The tensor network framework naturally generalizes beyond MPS to more complex network geometries. Notable examples include projected entangled pair states (PEPS) for two-dimensional lattices and tree or hierarchical tensor networks for systems with multiscale structure.

A unifying theoretical description of MPS and PEPS was developed by Cirac et al. [7], who analyzed their expressive power, entanglement properties, and variational capabilities. These developments established tensor networks as a general language for describing structured high-dimensional tensors with controlled complexity.

Orús [8] introduced tensor networks from a practical and algorithmic viewpoint, emphasizing diagrammatic notation, contraction strategies, and numerical stability. This work played a key role in disseminating tensor network methods beyond condensed matter physics.

1.2.4 Modern Viewpoint: Tensor Networks as Computational Models

In recent years, tensor networks have evolved from domain-specific tools into a general computational framework for high-dimensional problems. Orús [9] highlighted their role as efficient representations of complex quantum systems, while also emphasizing connections to machine learning, data compression, and scientific computing.

From a modern perspective, tensor networks can be viewed as structured low-rank models that balance expressiveness and computational tractability. Their success relies on exploiting locality, entanglement constraints, or hierarchical correlations in the underlying data or physical system.

Consequently, tensor networks now serve as a bridge between quantum physics, numerical analysis, and applied mathematics, providing scalable tools for problems that would otherwise be prohibitively high-dimensional.

1.3 Canonical Tensor Network Topologies and Their Restrictions

Tensor networks can be classified according to the geometry of their underlying graphs, which directly determines their inductive bias, expressive power, and computational cost. Different topologies impose different constraints on how correlations or entanglement can be represented and how efficiently contractions can be performed. This section reviews the most common tensor network structures used in practice.

1.3.1 Chain Tensor Networks: Matrix Product States and Tensor Trains

Chain tensor networks are characterized by a one-dimensional linear topology. The most prominent examples are Matrix Product States (MPS) and their numerical-analysis counterpart, the Tensor-Train (TT) decomposition.

From the physics perspective, MPS provide an efficient representation of quantum states in one-dimensional systems, where correlations are predominantly short-ranged. Schollwöck [6] showed that MPS naturally arise from the Density Matrix Renormalization Group and are particularly effective for states obeying an area law for entanglement entropy.

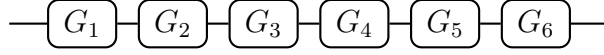
In applied mathematics, the same structure is known as the Tensor-Train format, introduced by Oseledets [4]. TT representations scale linearly with the tensor order and polynomially with the TT-ranks, making them attractive for high-dimensional numerical problems.

Inductive bias. Chain TNs assume a sequential ordering of modes and favor local correlations along the chain.

Contraction cost. Exact contraction is efficient and scales linearly in the number of tensors.

Limitations. Long-range correlations and multi-dimensional interaction patterns are difficult to capture without significantly increasing the ranks.

Chain tensor networks (Fig. 1.4) impose a sequential inductive bias: correlations are primarily mediated through neighboring cores, enabling efficient exact contractions



chain topology (MPS / TT)

Figure 1.4: Chain tensor network (MPS/TT): the tensor is factorized into a sequence of low-order cores connected by contracted “bond” indices; open legs correspond to physical/data indices.

and scalable algorithms.

1.3.2 Tree Tensor Networks: TTN and Hierarchical Tucker

Tree tensor networks generalize chain structures by organizing tensor modes in a hierarchical, tree-shaped topology. This class includes Tree Tensor Networks (TTN) and the Hierarchical Tucker (HT) decomposition.

Grasedyck [3] introduced the HT format as a mathematically rigorous extension of Tucker decomposition that avoids the exponential growth of the core tensor. Hackbusch [2] further developed the functional-analytic foundation of HT representations and their numerical properties.

Inductive bias. Tree TNs encode hierarchical or multilevel correlations and are well suited for problems with nested or multiscale structure.

Contraction cost. Contractions remain efficient due to the acyclic graph structure.

Limitations. The quality of approximation depends on the chosen dimension tree, which may require problem-specific design.

Tree tensor networks (Fig. 1.5) encode hierarchical correlations via a dimension tree, yielding favorable storage and stable truncations in hierarchical Tucker representations.

1.3.3 Two-Dimensional and Lattice Tensor Networks: PEPS

Projected Entangled Pair States (PEPS) extend MPS to higher-dimensional lattice geometries. They were introduced by Verstraete and Cirac [10] as a natural generalization for two-dimensional quantum systems.

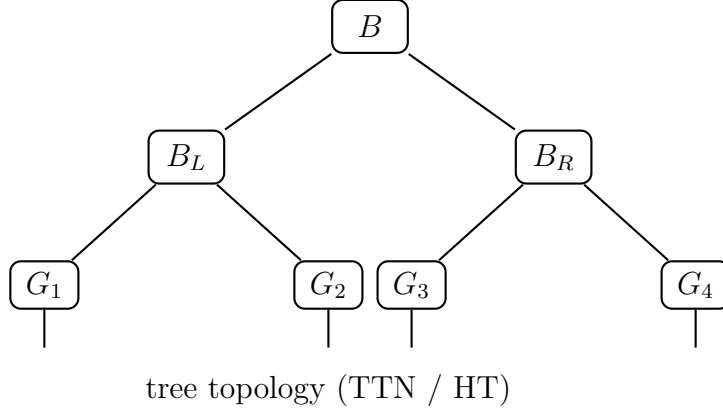


Figure 1.5: Tree tensor network (TTN/HT): a dimension tree organizes mode interactions hierarchically, reducing the effective core growth compared to Tucker and enabling efficient contractions due to the acyclic graph structure.

PEPS provide a powerful representation for states with genuine two-dimensional entanglement structure. Cirac et al. [7] analyzed PEPS within a unified tensor network framework, highlighting their expressive power and theoretical properties.

Inductive bias. PEPS encode locality in two-dimensional lattices and naturally model spatially local interactions.

Contraction cost. Exact contraction is computationally intractable in general and requires approximations.

Limitations. High computational cost limits PEPS applications to moderate system sizes.

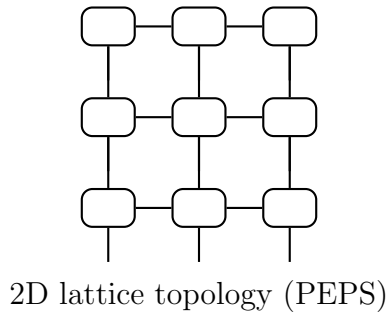
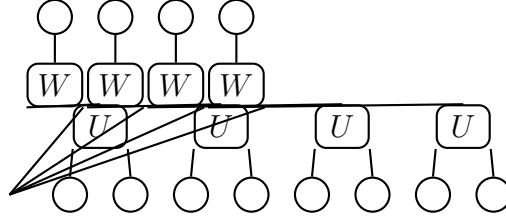


Figure 1.6: Projected entangled pair state (PEPS): tensors are arranged on a 2D lattice with local bonds; open legs represent physical indices. The 2D connectivity captures planar locality but makes exact contraction computationally demanding.

PEPS (Fig. 1.6) generalize MPS to two-dimensional lattices, providing a strong local-



multi-scale topology (MERA)

Figure 1.7: MERA (multiscale entanglement renormalization ansatz): alternating disentanglers (U) and isometries (W) implement a hierarchical coarse-graining, enabling efficient representation of scale-dependent and long-range correlations.

ity bias for 2D systems; however, their contraction typically requires approximations due to the high cost of exact evaluation.

1.3.4 Multi-Scale Tensor Networks: MERA

The Multiscale Entanglement Renormalization Ansatz (MERA) was introduced by Vidal [11] to represent quantum states with scale-invariant or critical behavior.

MERA incorporates explicit coarse-graining transformations and disentanglers, allowing it to efficiently capture long-range correlations.

Inductive bias. MERA assumes scale invariance and hierarchical renormalization structure.

Contraction cost. Contractions are efficient due to the causal cone structure.

Limitations. Implementation complexity is significantly higher than chain or tree TNs.

MERA (Fig. 1.7) introduces an explicit multiscale inductive bias through disentanglers and isometries, making it effective for critical or scale-invariant systems while keeping contractions tractable via the causal-cone structure.

1.3.5 Cyclic Tensor Networks: Tensor Ring

Cyclic tensor networks remove the boundary conditions of chain TNs by forming a closed-loop topology. The Tensor Ring (TR) decomposition was introduced by Zhao

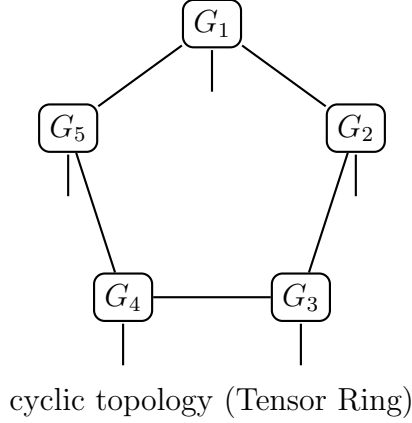


Figure 1.8: Tensor Ring (TR): a cyclic generalization of TT where the cores form a closed loop. The topology removes boundary effects and treats modes more symmetrically, but cyclicity complicates contraction and optimization.

et al. [12] as a cyclic generalization of the Tensor-Train format.

Inductive bias. Tensor Ring representations eliminate artificial boundary effects and treat all modes symmetrically.

Contraction cost. Contractions are more expensive than TT due to cyclic dependencies.

Limitations. Optimization and rank selection are more challenging than in acyclic networks.

Tensor Ring models (Fig. 1.8) replace the TT chain by a cycle, reducing boundary artifacts and improving symmetry across modes; the resulting cyclic dependencies can increase contraction and training complexity.

1.3.6 Summary of Topological Trade-Offs

Each tensor network topology represents a trade-off between expressiveness, computational efficiency, and inductive bias. Chain and tree TNs offer efficient contraction and scalability, while lattice and multiscale TNs provide greater expressive power at increased computational cost.

1.4 Where Tensor Networks Sit in Modern AI and Machine Learning

In recent years, tensor networks have emerged as a bridge between ideas from quantum many-body physics, numerical analysis, and modern artificial intelligence. Their relevance to machine learning stems from two complementary perspectives: tensor networks as efficient parametric models and tensor networks as structured probabilistic representations. In both cases, the underlying network topology plays a critical role in determining expressive power, inductive bias, and computational cost.

1.4.1 Tensor Networks as Efficient Models

One of the earliest connections between tensor networks and supervised learning was established by Stoudenmire and Schwab [13], who demonstrated that matrix product states can be used as classifiers for high-dimensional data. In this setting, the input features are mapped to local feature vectors and contracted with an MPS, yielding a scalar output corresponding to a class score. Despite their relatively small number of parameters, such models were shown to achieve competitive performance on standard benchmarks.

A key motivation for using tensor networks in deep learning is parameter efficiency. Novikov et al. [14] introduced tensor-train (TT) factorizations of fully connected layers, significantly reducing the number of trainable parameters while preserving accuracy. This line of work, often referred to as *tensorizing neural networks*, exploits the same low-rank structure that underlies successful tensor decompositions in scientific computing.

From this viewpoint, tensor networks act as structured replacements for dense weight tensors. Their topology encodes an inductive bias on how input dimensions interact, enabling strong compression without a proportional loss in expressive power. Chain and tree topologies, in particular, have proven effective in controlling model complexity and improving generalization.

1.4.2 Tensor Networks as Probabilistic Representations

Beyond deterministic models, tensor networks also provide a powerful language for probabilistic modeling. Glasser et al. [15] analyzed the expressive power of tensor-network factorizations for representing probability distributions, showing that

different network topologies correspond to distinct families of probabilistic models with varying representational capacity.

In this context, tensor networks can be interpreted as factorizations of high-dimensional probability tensors, analogous to how graphical models factorize joint probability distributions. Robeva and Seigal [16] formalized this connection by establishing a duality between tensor networks and graphical models. Their work shows that conditional independence structures in probabilistic models correspond directly to network topology and contraction structure in tensor networks.

This probabilistic interpretation clarifies why topology matters: the connectivity of the network determines which correlations can be efficiently represented and which require high ranks. For example, chain tensor networks impose a sequential dependency structure, while tree or lattice topologies allow for more complex conditional dependencies.

1.4.3 The Role of Topology and Inductive Bias

Across both deterministic and probabilistic perspectives, the choice of tensor network topology acts as a form of inductive bias. It constrains the class of functions or distributions that can be represented efficiently, thereby influencing learning dynamics, sample efficiency, and generalization behavior.

From a machine learning standpoint, tensor networks occupy an intermediate position between fully connected neural networks and classical graphical models. They offer a principled way to balance expressiveness and computational tractability by explicitly encoding structural assumptions through topology.

As a result, tensor networks are increasingly viewed not merely as compression tools, but as model classes in their own right, with topology serving as a key design parameter in modern AI and machine learning systems.

1.5 Thesis Problem Statement and Contributions

The preceding sections have established tensor networks (TNs) as powerful tools for representing and manipulating high-dimensional objects, with applications ranging from numerical analysis and quantum many-body physics to modern machine learning. Across these domains, a recurring theme is that the *topology* of a tensor network plays a central role in determining its expressive power, computational cost, and

inductive bias.

Despite their success, most tensor network methods rely on a small set of fixed, canonical topologies—such as chains (MPS/TT), trees (TTN/HT), grids (PEPS), or multiscale constructions (MERA). While these structures are well motivated for specific classes of problems, they impose strong and often implicit assumptions about the underlying correlation structure of the data. As a result, a mismatch between network topology and problem structure can lead to inefficient representations, unnecessarily large ranks, or poor generalization.

From the perspective of machine learning, this limitation mirrors the broader issue of hand-designed model architectures. As discussed in Section 1.4, tensor networks can be interpreted as structured parametric models whose topology encodes an inductive bias. Fixing this topology a priori restricts the hypothesis space and may prevent the model from adapting to problem-specific interaction patterns. In high-dimensional settings, where the true structure is rarely known in advance, such rigidity becomes a significant bottleneck.

These observations motivate the central problem addressed in this thesis: *how to move beyond fixed tensor network topologies and enable adaptive or data-driven structure selection*. Rather than treating topology as a static design choice, this work views it as a variable to be optimized, searched, or learned in conjunction with tensor parameters.

1.5.1 Thesis Contributions

The main contributions of this thesis are summarized as follows:

- A unified perspective on tensor network topology as an explicit inductive bias governing expressiveness, efficiency, and generalization in high-dimensional models.
- An analysis of the limitations of canonical tensor network structures when applied to problems with unknown or nontrivial correlation patterns.
- The development and investigation of methods for tensor network structure and topology search, enabling adaptive representations tailored to the underlying data or task.
- Empirical and theoretical insights demonstrating how learned or optimized topologies can outperform fixed architectures in terms of compression efficiency

and predictive performance.

1.5.2 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 reviews existing approaches to tensor network structure search and related methods from machine learning and numerical optimization. Subsequent chapters introduce the proposed methodologies, analyze their theoretical properties, and evaluate their performance on representative benchmark problems.

By treating tensor network topology as a learnable or optimizable object rather than a fixed design choice, this thesis aims to expand the applicability of tensor networks and provide a principled framework for adaptive high-dimensional modeling.

Chapter 2

Related Work

2.1 Formulating Tensor Network Topology Search

A central challenge in tensor network (TN) methods is the selection of an appropriate network topology. As established in Chapter 1, the topology of a tensor network determines both its expressive power and the computational cost of fundamental operations such as contraction. This section reviews how topology selection can be formulated as a search problem and why it is inherently difficult.

2.1.1 Topology as a Discrete Structural Variable

From a mathematical perspective, a tensor network is defined by two components: the local tensors and the pattern of index contractions connecting them. While tensor entries are continuous parameters that can be optimized using gradient-based methods, the network topology itself is a discrete object, typically represented as a graph.

Selecting a tensor network topology therefore amounts to choosing a graph structure that specifies how tensor indices are connected. This choice determines which correlations can be represented efficiently and how intermediate tensor sizes grow during contraction. Unlike rank selection, which can often be handled through continuous relaxation or truncation, topology selection involves a combinatorial search over possible graph structures.

2.1.2 Topology and Contraction Complexity

The computational cost of contracting a tensor network depends critically on the order in which tensor contractions are performed and on the structure of the underlying graph. Markov and Shi [17] provided a foundational analysis linking tensor network contraction complexity to graph-theoretic notions such as treewidth. They showed that the cost of contracting a general tensor network scales exponentially with the treewidth of its underlying graph, establishing a direct connection between topology and computational tractability.

This result has two important implications. First, even when a tensor network represents a compact low-rank model, poor topological choices can render exact contraction infeasible. Second, finding an optimal contraction strategy—or equivalently, an optimal topology—is itself computationally hard, as treewidth minimization is an NP-hard problem.

Consequently, canonical tensor network topologies such as chains or trees are popular not only because of their expressive properties, but also because they guarantee bounded treewidth and efficient contraction. More expressive topologies, such as grids or general graphs, often require approximate contraction schemes to remain computationally viable.

2.1.3 Expensive Evaluation and Search Challenges

Topology search is further complicated by the cost of evaluating candidate structures. Assessing the quality of a tensor network topology typically requires repeated contractions, rank adaptations, or full model training, all of which are computationally expensive.

In practice, contraction order optimization and topology selection are often intertwined. Tools such as `opt_einsum` [18] address part of this challenge by automatically identifying efficient contraction paths for a fixed tensor network structure. However, these methods do not modify the underlying topology and therefore cannot overcome limitations imposed by an unfavorable graph structure.

As a result, most existing approaches rely on heuristics, restricted topology families, or problem-specific insights to limit the search space. While effective in certain settings, these strategies do not provide a general solution to the problem of adaptive tensor network topology selection.

Therefore, tensor network topology search can be viewed as a discrete, combinatorial

optimization problem characterized by expensive evaluation and strong coupling between expressiveness and computational cost. Theoretical results linking contraction complexity to graph properties highlight the importance of structure, while practical considerations limit exhaustive exploration of the topology space.

These challenges motivate the development of principled methods for restricting, guiding, or approximating topology search, which will be reviewed in the remainder of this chapter.

2.2 Evolutionary and Heuristic Topology Search

Given the combinatorial nature of tensor network (TN) topology selection and the high cost of evaluating candidate structures, several works have explored heuristic and evolutionary approaches to topology search. These methods aim to navigate the discrete space of network structures using population-based optimization, avoiding exhaustive enumeration while still allowing for flexible architectures.

2.2.1 Evolutionary Formulation

Li et al. [19] proposed one of the first explicit evolutionary approaches for tensor network topology search. In their framework, a tensor network structure is encoded as a graph-based genome, where nodes correspond to tensors and edges represent contracted indices. Evolutionary operators such as mutation and crossover are used to generate new candidate topologies from an existing population.

Each candidate topology is evaluated by performing tensor network decomposition or approximation on a target dataset, and its fitness is measured in terms of reconstruction error and model complexity. Through successive generations, the population evolves toward architectures that better balance approximation accuracy and computational efficiency.

This approach is appealing because it treats topology as a first-class optimization variable and does not restrict the search to a predefined family of canonical structures.

2.2.2 Heuristic Search Characteristics

Evolutionary topology search methods rely on heuristic guidance rather than explicit theoretical guarantees. Their effectiveness depends on carefully designed mutation

rules, selection strategies, and fitness functions. In practice, the search space is often constrained to ensure tractability, for example by limiting the maximum degree of nodes or restricting candidate graphs to remain connected and contractible.

A key advantage of such heuristics is their flexibility: they can discover unconventional topologies that outperform standard chain or tree architectures on specific tasks. However, this flexibility comes at the cost of significant computational overhead, as each candidate topology must be evaluated through expensive tensor contractions or decompositions.

2.2.3 Limitations and Open Challenges

Despite their conceptual appeal, evolutionary and heuristic topology search methods face several fundamental limitations. First, the cost of evaluating candidate structures severely limits scalability, especially for high-order tensors or large datasets. Second, the stochastic nature of evolutionary algorithms can lead to unstable or irreproducible results, making systematic analysis difficult.

Moreover, these approaches typically decouple topology search from tensor optimization: for each candidate structure, tensor parameters must be reinitialized or retrained, which further increases computational cost. As a result, evolutionary methods are often restricted to small-scale problems or used as proof-of-concept demonstrations rather than as general-purpose solutions.

2.2.4 Positioning Within the State of the Art

Evolutionary and heuristic topology search methods represent an important step toward adaptive tensor network models. They demonstrate that relaxing the assumption of fixed topology can lead to improved performance and more efficient representations. At the same time, their limitations highlight the need for more principled and scalable approaches that integrate topology selection with efficient optimization and contraction strategies.

These observations motivate subsequent research directions that seek to combine structural flexibility with computational tractability, which will be discussed in the following sections.

2.3 Local and Permutation-Based Structure Search

While evolutionary approaches treat tensor network (TN) topology as a global and largely unconstrained optimization variable, an alternative line of work focuses on *local* or *permutation-based* structure search. These methods exploit the observation that, for certain tensor network families—most notably chain-based architectures such as Tensor Trains (TT) or Matrix Product States (MPS)—the topology is fixed up to a permutation of tensor modes. Optimizing this permutation can substantially improve approximation quality and computational efficiency without altering the underlying network geometry.

2.3.1 Permutation Search as a Structural Optimization Problem

In chain tensor networks, the ordering of tensor modes determines which dimensions are grouped together early in the contraction process and which interactions are mediated through long sequences of low-rank cores. As a result, different permutations of the same modes can lead to dramatically different rank profiles, storage costs, and approximation errors.

Li et al. [20] formalized this observation by framing mode permutation as a discrete local search problem. Starting from an initial ordering, their method iteratively applies local permutations—such as swaps of neighboring modes—and evaluates the impact on tensor network compression or reconstruction error. By restricting the search to local moves, the method avoids the combinatorial explosion associated with global topology search.

2.3.2 Efficiency and Practical Advantages

A key advantage of permutation-based approaches is their relatively low computational overhead. Because the underlying TN topology remains unchanged, existing contraction strategies and optimization routines can be reused. Moreover, local updates allow for incremental evaluation, making it possible to explore a large number of candidate permutations at a fraction of the cost required by evolutionary methods.

From a practical standpoint, these methods strike a balance between adaptivity and tractability. They retain the favorable contraction properties of chain tensor networks while introducing a limited but meaningful degree of structural flexibility.

2.3.3 Limitations of Permutation-Based Search

Despite their efficiency, permutation-based methods are inherently constrained by the assumption of a fixed network topology. They cannot introduce new connections, remove existing ones, or change the overall graph structure of the tensor network. Consequently, their expressive power is bounded by the inductive bias of the chosen topology, typically that of a one-dimensional chain.

In problems where correlations are not well aligned with any linear ordering of modes, permutation-based optimization may yield only marginal improvements. Furthermore, local search strategies can become trapped in suboptimal configurations, particularly when the objective landscape is highly nonconvex.

2.3.4 Positioning Within the State of the Art

Local and permutation-based structure search methods occupy an intermediate position between fixed-topology approaches and fully adaptive topology search. They demonstrate that even modest structural flexibility can lead to significant gains in performance, while also highlighting the limitations of approaches that operate within a single topological family.

These methods provide important insights into the role of structure in tensor network models and motivate the exploration of more general frameworks that combine local efficiency with global structural adaptability, which will be discussed in the subsequent sections.

2.4 Alternating and Neighborhood Exploration Methods

An intermediate class of tensor network (TN) structure search methods lies between fully global topology optimization and strictly local permutation-based approaches. These methods explore a *restricted neighborhood* of candidate structures while alternating between structural updates and parameter optimization. As such, they are particularly relevant to practical implementations and are closest in spirit to the approach developed in this thesis.

2.4.1 Tensor Network Architecture Learning (TnALE)

Li et al. [21] introduced Tensor Network Architecture Learning (TnALE), a framework that explicitly alternates between tensor parameter optimization and local architecture modifications. Rather than searching over arbitrary graph structures, TnALE defines a neighborhood around a current tensor network topology and explores this neighborhood through a sequence of controlled structural operations.

Typical neighborhood moves include local edge rewiring, rank redistribution, or substructure replacement, all designed to preserve contractibility and avoid the exponential complexity associated with unrestricted topology changes. After each structural modification, tensor parameters are re-optimized to evaluate the quality of the updated architecture.

This alternating strategy allows TnALE to adapt network structure in response to data while maintaining computational tractability.

2.4.2 Advantages of Alternating Structure–Parameter Optimization

The key advantage of neighborhood-based alternating methods is their balance between expressive flexibility and computational efficiency. By restricting the search space to a local neighborhood, these approaches avoid the combinatorial explosion inherent in global topology search while still enabling meaningful structural adaptation.

Moreover, alternating optimization aligns well with practical training pipelines: structural changes are interleaved with standard tensor optimization routines, allowing reuse of existing contraction strategies and optimization algorithms. This design makes such methods significantly more scalable than evolutionary approaches and more flexible than purely permutation-based methods.

2.4.3 Limitations and Open Issues

Despite their practical appeal, alternating and neighborhood exploration methods remain constrained by the definition of the neighborhood itself. The choice of allowable structural moves implicitly encodes a strong inductive bias and may prevent the discovery of qualitatively different architectures.

Additionally, frequent re-optimization of tensor parameters after structural updates

can still be computationally expensive, particularly for large-scale problems. As with other local search strategies, there is also a risk of convergence to locally optimal architectures that are suboptimal from a global perspective.

2.4.4 Positioning Within the State of the Art

Alternating neighborhood exploration methods such as TnALE represent an important step toward adaptive tensor network architectures. They demonstrate that structure learning can be integrated into tensor network optimization pipelines without incurring prohibitive computational cost.

At the same time, their reliance on predefined neighborhoods highlights the need for more general frameworks that allow broader structural adaptation while retaining efficiency. These considerations directly motivate the approach proposed in this thesis, which seeks to expand the space of accessible topologies while preserving practical tractability.

2.5 Bayesian Structure Search and Uncertainty-Aware Selection

A recent and conceptually distinct line of work frames tensor network (TN) topology search as a Bayesian inference problem. Rather than identifying a single optimal structure through heuristic or local exploration, these approaches aim to model uncertainty over the space of possible topologies and to guide search using probabilistic surrogate models. This perspective is particularly well suited to settings where structure evaluation is expensive and noisy.

2.5.1 Bayesian Tensor Network Structure Search

Zeng et al. [22] proposed one of the first Bayesian formulations of tensor network structure search. In their approach, the choice of TN topology is treated as a latent variable, and a posterior distribution over candidate structures is inferred based on observed performance metrics, such as approximation error or predictive accuracy.

By incorporating prior assumptions and probabilistic uncertainty, this framework enables principled exploration–exploitation trade-offs during search. Promising structures can be refined, while uncertainty-aware selection helps avoid premature

commitment to suboptimal architectures. This Bayesian viewpoint contrasts with deterministic or purely heuristic methods by explicitly quantifying confidence in structural decisions.

2.5.2 Surrogate-Based Search with Gaussian Processes

To address the high computational cost of evaluating tensor network structures, Zeng et al. [23] introduced *tnGPS*, a surrogate-assisted structure search framework based on Gaussian processes. In *tnGPS*, a Gaussian process model is trained to approximate the mapping from tensor network structure descriptors to performance metrics. This surrogate is then used to guide the exploration of the topology space through Bayesian optimization.

By decoupling expensive tensor contractions from the search process, *tnGPS* significantly reduces the number of full evaluations required. Uncertainty estimates provided by the surrogate model play a central role in selecting which candidate structures to evaluate next, prioritizing those with high expected improvement.

2.5.3 Advantages of Uncertainty-Aware Structure Search

Bayesian and surrogate-based approaches offer several advantages over earlier methods. First, they provide a systematic mechanism for balancing exploration of novel topologies against exploitation of known good structures. Second, uncertainty estimates enable more sample-efficient search, which is crucial when each evaluation involves costly tensor optimization or contraction.

From a methodological standpoint, these approaches elevate tensor network structure search from heuristic optimization to a statistically grounded inference problem.

2.5.4 Limitations and Open Challenges

Despite their conceptual appeal, Bayesian structure search methods face several challenges. The definition of suitable structure encodings for surrogate models remains nontrivial, particularly for complex or heterogeneous tensor network topologies. Moreover, surrogate accuracy may degrade as the search space grows, limiting scalability to very large or highly expressive architectures.

Additionally, Bayesian methods typically introduce substantial algorithmic and implementation complexity, which may hinder adoption in large-scale or real-time

applications. As with other structure search approaches, the choice of priors and acquisition strategies can strongly influence outcomes.

2.5.5 Positioning Within the State of the Art

Bayesian and uncertainty-aware methods represent the most recent and theoretically grounded direction in tensor network topology search. They complement evolutionary, local, and neighborhood-based approaches by explicitly modeling uncertainty and by improving sample efficiency.

At the same time, their reliance on surrogate models and predefined structure encodings highlights the ongoing trade-off between generality, scalability, and practical usability. These observations further motivate the need for structure-adaptive methods that combine efficient local exploration with principled decision-making, as pursued in this thesis.

2.6 Adaptive and Greedy Tensor Network Structure Learning

A complementary approach to tensor network (TN) topology search focuses on adaptive, greedy strategies that incrementally construct network structure based on local improvement criteria. Rather than exploring a large space of candidate architectures or relying on surrogate models, these methods build tensor networks progressively by identifying and refining structural components that most effectively reduce a given objective.

A representative example of this paradigm is the greedy structure learning framework proposed by Hashemizadeh et al. [24], which introduces an adaptive algorithm for jointly learning TN structure and parameters from data.

2.6.1 Greedy Structure Construction

The method proposed in [24] formulates tensor network learning as a bi-level optimization problem. At the upper level, the TN structure is selected under a constraint on the total number of parameters, while at the lower level, tensor parameters are optimized with respect to a task-specific loss function. Directly solving this problem is intractable due to the combinatorial nature of the structure space.

To address this challenge, the authors propose a greedy algorithm that starts from a rank-one tensor network and iteratively expands the structure. At each iteration, the algorithm identifies the most promising edge whose rank should be increased, based on its expected impact on the loss. This local structural modification is followed by continuous optimization of tensor parameters, with careful weight transfer to preserve information from previous iterations.

In addition to rank adaptation, the method allows for the creation of internal nodes through node-splitting operations, enabling the discovery of nontrivial topologies beyond simple chain or ring structures.

2.6.2 Strengths of the Greedy Approach

A key strength of greedy adaptive structure learning is its tight integration of structure selection and parameter optimization. By reusing optimized parameters across iterations, the method avoids the costly re-initialization required by evolutionary or population-based approaches. This leads to substantial computational savings while still allowing the network structure to adapt to data.

Empirical results reported in [24] demonstrate that the greedy algorithm can outperform fixed-topology methods such as CP, Tucker, TT, and TR, as well as evolutionary topology search, across tasks including tensor decomposition, tensor completion, and neural network compression. Notably, the method achieves these improvements with significantly fewer parameters and lower computational cost.

2.6.3 Limitations and Relation to Other Methods

Despite its effectiveness, greedy structure learning remains inherently local. The sequence of structural decisions depends on earlier choices, which may lead to convergence toward locally optimal tensor network structures. Moreover, the greedy criteria used to select edges or node splits introduce implicit heuristics that may bias the resulting architectures.

Compared to Bayesian and surrogate-based approaches reviewed in Section 2.5, greedy methods do not explicitly model uncertainty over structures. However, they offer a substantially lower algorithmic and computational overhead, making them attractive for large-scale or practical applications.

2.6.4 Positioning Within the State of the Art

Adaptive greedy tensor network structure learning represents a pragmatic and scalable alternative to global topology search. It occupies a middle ground between rigid fixed-topology models and fully probabilistic structure inference, combining efficiency with meaningful structural flexibility.

This line of work is particularly relevant to the approach developed in this thesis, as it demonstrates that effective structure adaptation can be achieved through incremental, locality-driven updates. At the same time, its limitations highlight opportunities for extending greedy mechanisms toward more flexible or principled structure exploration strategies, which motivates the contributions presented in the next chapter.

2.7 Limitations of Prior Work and Scope of This Thesis

The review in Sections 2.1–2.6 highlights significant progress in tensor network (TN) topology and structure search. At the same time, existing approaches share several fundamental limitations that constrain their scalability, robustness, and applicability in practical settings. This section summarizes these limitations and clarifies how they motivate the problem formulation adopted in this thesis.

2.7.1 High Evaluation Cost of Structure Search

A common challenge across nearly all structure search methods is the high cost of evaluating candidate tensor network architectures. For a given topology, evaluation typically requires full or partial optimization of tensor parameters, followed by repeated tensor contractions. As a result, the overall computational cost scales multiplicatively with the number of candidate structures and the cost of training each candidate.

This issue is particularly pronounced in evolutionary, Bayesian, and surrogate-based methods, where a large number of candidate topologies must be explored to achieve reliable performance estimates. Even greedy and neighborhood-based approaches incur substantial overhead due to frequent retraining after structural modifications. As observed in adaptive and greedy methods [24], this cost often dominates total runtime and limits applicability to small-scale problems.

2.7.2 Late Enforcement of Feasibility Constraints

Another limitation of existing work is that feasibility constraints are often enforced only after candidate structures have been generated. Constraints related to contraction cost, memory usage, or intermediate tensor sizes are frequently treated as secondary considerations or addressed through ad hoc pruning.

However, as established by the contraction complexity analysis of Markov and Shi [17], tensor network contraction cost depends critically on structural properties such as node degrees and graph treewidth. Failure to account for these constraints early in the search process can lead to catastrophic cost explosions, rendering otherwise promising architectures infeasible.

This disconnect between structure generation and feasibility assessment motivates search strategies that integrate cost-awareness directly into the objective or constraint formulation.

2.7.3 Lack of Explicit Multiobjective Optimization

A further limitation of prior work is the absence of explicit multiobjective optimization. Most existing methods optimize a single scalar objective, typically reconstruction error or predictive accuracy, while treating other criteria—such as compression ratio, runtime, or perceptual quality—as secondary or implicit considerations.

In practice, tensor network design involves inherent trade-offs between competing objectives. For example, improved accuracy may require increased ranks and parameters, while higher compression can degrade perceptual quality. These trade-offs are rarely modeled explicitly, leading to architectures that are optimal under one metric but suboptimal under others.

2.7.4 Motivation for Min–Max Scalarization

To address this limitation, this thesis adopts a min–max scalarization framework that treats tensor network structure selection as a multiobjective optimization problem. Rather than collapsing multiple criteria into a weighted sum, the proposed approach minimizes the maximum normalized deviation across a set of competing objectives, including reconstruction error, compression ratio, and perceptual quality metrics.

This formulation enables balanced optimization and prevents any single objective from dominating the search process. Importantly, feasibility constraints on model

size and estimated contraction cost are enforced *prior* to evaluation, ensuring that only tractable architectures are considered. The resulting search procedure integrates structural feasibility, performance, and compression into a unified decision criterion. The practical realization of this framework is implemented through a local neighborhood exploration strategy with explicit safety constraints and min–max objective evaluation, as detailed in the accompanying implementation .

2.7.5 Positioning of This Thesis

In summary, prior work on tensor network structure search is limited by high evaluation costs, delayed feasibility enforcement, and insufficient treatment of multiobjective trade-offs. This thesis addresses these limitations by introducing a cost-aware, multi-objective structure search framework that balances expressiveness, efficiency, and reconstruction quality within a principled optimization setting.

The next chapter presents the proposed method in detail, including its objective formulation, constraint handling, and optimization strategy.

Chapter 3

Proposed methodology

3.1 Problem Setup: Image Tensorization and Reconstruction Objective

This thesis considers the problem of learning tensor network (TN) representations for image data under explicit structural and computational constraints. The goal is to identify a tensor network structure and parameters that enable accurate image reconstruction while balancing compression efficiency and computational feasibility.

3.1.1 Image Tensorization

Let $X \in \mathbb{R}^{H \times W}$ denote a grayscale image of fixed resolution. To enable tensor network modeling, the image is first mapped to a higher-order tensor through a tensorization operation

$$\mathcal{X} = \mathcal{T}(X) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N},$$

where $\prod_{n=1}^N I_n = HW$ and $\mathcal{T}$ denotes a deterministic reshaping or reindexing operator.

Two tensorization strategies are considered in this work. The first is a direct reshape-based tensorization, which maps the flattened image vector into an N -way tensor with equal mode sizes, preserving linear adjacency in memory order. This approach is simple and computationally efficient but does not explicitly encode spatial locality.

The second strategy employs a structured reindexing based on a base-4 digit expansion of pixel coordinates. In this case, spatial indices are reordered according to a Morton-like (Z-order) space-filling curve, which interleaves the binary representations of row

and column indices. This mapping preserves spatial locality across tensor modes more effectively than linear reshaping and has been widely used in multidimensional indexing and data locality optimization [25].

Both tensorization schemes yield tensors of identical shape but induce different correlation structures across modes, which can significantly affect the efficiency and expressiveness of subsequent tensor network factorizations.

3.1.2 Tensor Network Reconstruction Model

Given a tensorized image $\mathcal{X}$, the objective is to approximate it using a tensor network model $\hat{\mathcal{X}}(\theta, \mathcal{G})$, where $\mathcal{G}$ denotes the network topology and θ represents the collection of tensor parameters. The approximation is obtained by contracting all internal indices of the network, producing a full tensor of the same shape as $\mathcal{X}$.

The primary reconstruction objective is defined in terms of relative squared error:

$$\mathcal{L}_{\text{rec}}(\theta, \mathcal{G}) = \frac{\|\mathcal{X} - \hat{\mathcal{X}}(\theta, \mathcal{G})\|_F^2}{\|\mathcal{X}\|_F^2}.$$

This loss function is scale-invariant and provides a natural measure of fidelity for tensor approximation tasks.

After reconstruction, the inverse tensorization operator $\mathcal{T}^{-1}$ is applied to obtain an image $\hat{X} = \mathcal{T}^{-1}(\hat{\mathcal{X}})$ in the original pixel domain. This allows evaluation using both tensor-based metrics and image-level quality measures, such as peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

3.1.3 Design Implications

The choice of tensorization directly influences the inductive bias imposed on the tensor network. Reshape-based tensorization favors global correlations aligned with memory order, while Morton-like tensorization promotes locality-aware interactions across tensor modes. As a result, tensorization and network topology must be considered jointly when designing efficient tensor network models for image reconstruction.

This problem formulation establishes the foundation for the min-max, cost-aware structure search framework introduced in the following sections.

3.2 Model: Arbitrary-Graph Tensor Network Parameterization

This section introduces the tensor network (TN) model class used throughout this thesis. Unlike canonical architectures with fixed topologies, the proposed model allows arbitrary graph structures, enabling fine-grained control over expressivity and computational cost. The model is defined by a set of tensor cores associated with graph nodes and a set of contracted indices associated with graph edges.

3.2.1 Graph-Based Parameterization

Let $G = (V, E)$ be an undirected graph with $|V| = N$ nodes. Each node $i \in V$ corresponds to a tensor core

$$\mathcal{G}_i \in \mathbb{R}^{I_i \times r^{\deg(i)}},$$

where I_i is the physical dimension associated with node i and $\deg(i)$ denotes the degree of the node in the graph. Each edge $(i, j) \in E$ introduces a shared bond index of dimension r between $\mathcal{G}_i$ and $\mathcal{G}_j$.

The full tensor represented by the network is obtained by contracting all bond indices:

$$\hat{\mathcal{X}} = \text{Contract}(\{\mathcal{G}_i\}_{i=1}^N, E),$$

yielding a tensor in $\mathbb{R}^{I_1 \times \dots \times I_N}$. This construction generalizes classical tensor formats such as TT, HT, and TR by allowing arbitrary connectivity patterns.

3.2.2 Binary Encoding of Topology

To enable algorithmic structure search, the network topology is encoded as a binary vector

$$\mathbf{b} \in \{0, 1\}^{N(N-1)/2},$$

where each entry corresponds to the presence or absence of a potential edge between a pair of nodes. This encoding establishes a one-to-one correspondence between bit vectors and graph structures, allowing local topology modifications to be implemented as bit flips.

Given a binary vector $\mathbf{b}$, the corresponding edge set $E(\mathbf{b})$ is constructed deterministically. This representation enables efficient neighborhood exploration while

maintaining a clear separation between discrete structural variables and continuous tensor parameters.

3.2.3 Einsum-Based Contraction Representation

For a fixed topology, contraction of the tensor network is implemented using Einstein summation notation. Each tensor core is assigned a label for its physical index and labels for each of its incident bond indices. The global contraction is then expressed as a single einsum equation of the form

$$\hat{\mathcal{X}} = \text{einsum}(\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_N),$$

where repeated labels denote contracted indices and uncontracted labels correspond to physical modes.

This representation provides a flexible and implementation-independent description of arbitrary tensor network contractions. Moreover, it allows the reuse of optimized contraction planning tools while keeping the model definition independent of contraction order.

3.2.4 Expressivity and Computational Trade-Offs

Allowing arbitrary graph structures significantly increases the expressive power of the tensor network. Dense or highly connected graphs can capture complex, nonlocal correlations that are inaccessible to chain or tree architectures. From a probabilistic perspective, this expressivity corresponds to modeling higher-order dependencies and conditional interactions among variables.

However, increased expressivity comes at the cost of computational tractability. As established by the connection between tensor network contraction and probabilistic graphical model marginalization [16], the complexity of contraction depends critically on graph structure. Highly connected graphs induce large intermediate tensors and exponential growth in contraction cost.

Miller et al. [26] further formalized this connection by framing tensor networks as a unifying representation for probabilistic graphical models. In this view, graph connectivity controls both representational power and the complexity of marginalization, reinforcing the need for explicit structure control.

3.2.5 Design Implications

The arbitrary-graph parameterization adopted in this thesis provides a unified model class that subsumes many existing tensor network architectures. At the same time, it necessitates careful management of structural complexity to avoid infeasible contractions. This tension between expressivity and tractability motivates the cost-aware and constraint-driven structure search framework developed in the subsequent sections.

3.3 Evaluation Metrics: Reconstruction, Compression, and Perceptual Quality

The performance of a tensor network (TN) representation cannot be adequately characterized by a single scalar measure. In image reconstruction tasks, high fidelity, compact representation, and perceptual quality often conflict. To capture these trade-offs explicitly, this thesis evaluates candidate tensor network structures using a set of complementary metrics spanning reconstruction error, compression efficiency, and perceptual similarity.

3.3.1 Relative Squared Error (RSE)

The primary reconstruction metric used in this work is the relative squared error (RSE), defined as

$$\text{RSE} = \frac{\|\mathcal{X} - \hat{\mathcal{X}}\|_F^2}{\|\mathcal{X}\|_F^2},$$

where $\mathcal{X}$ denotes the tensorized input image and $\hat{\mathcal{X}}$ its tensor network reconstruction.

RSE is scale-invariant and directly measures approximation fidelity in the tensor domain. It is widely used in tensor decomposition and compression literature due to its simplicity and strong theoretical grounding. However, RSE does not necessarily correlate with perceptual image quality, motivating the use of additional image-level metrics.

3.3.2 Compression Ratio

Compression efficiency is quantified through the effective compression ratio, defined as:

$$\text{CR} = \frac{\text{number of elements in the original image}}{\text{number of trainable TN parameters}}.$$

This metric reflects the storage efficiency of the tensor network representation and directly captures the trade-off between model complexity and reconstruction accuracy.

In contrast to rank-based measures alone, the parameter-count formulation allows fair comparison across different network topologies and tensorizations.

3.3.3 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is a standard image quality metric derived from the mean squared error (MSE) between the original image X and its reconstruction $\hat{X}$:

$$\text{PSNR} = 10 \log_{10} \left(\frac{L^2}{\text{MSE}} \right), \quad \text{MSE} = \frac{1}{HW} \sum_{i,j} (X_{ij} - \hat{X}_{ij})^2,$$

where L denotes the maximum possible pixel value.

PSNR is widely used in image compression and transmission standards due to its simplicity and reproducibility. Standardized implementations and evaluation protocols are described in technical recommendations from telecommunications agencies [27]. Despite its popularity, PSNR is known to correlate imperfectly with human visual perception, particularly for structured distortions.

3.3.4 Structural Similarity Index (SSIM)

The Structural Similarity Index Measure (SSIM) is a perceptually motivated image quality metric designed to model human visual perception more accurately than pixel-wise error measures. SSIM was introduced by Wang et al. [28] and is based on the observation that the human visual system is highly adapted to extract structural information from images.

Given two image patches x and y , SSIM is defined as the product of three comparison functions measuring luminance, contrast, and structural similarity:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma,$$

where α , β , and γ are nonnegative exponents typically set to 1.

The individual components are defined as follows:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3},$$

where μ_x and μ_y denote local mean intensities, σ_x^2 and σ_y^2 local variances, and σ_{xy} the local covariance between x and y . The constants C_1 , C_2 , and C_3 are small positive values introduced to stabilize the division when denominators are close to zero.

In the commonly used simplified form with $C_3 = C_2/2$ and $\alpha = \beta = \gamma = 1$, SSIM reduces to:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}.$$

For full images, SSIM is computed locally using a sliding window and averaged over all windows to obtain a global similarity score. The resulting value typically lies in the interval $[-1, 1]$, with higher values indicating greater perceptual similarity.

Unlike RSE or PSNR, SSIM explicitly accounts for structural distortions, making it particularly suitable for evaluating reconstructed images produced by compressed tensor network representations.

3.3.5 Multiobjective Perspective

Each metric captures a distinct aspect of tensor network performance: RSE measures tensor-level fidelity, compression ratio reflects model efficiency, and PSNR/SSIM assess perceptual image quality. Optimizing any single metric in isolation may lead to undesirable outcomes, such as high-fidelity but inefficient models or highly compressed representations with poor visual quality.

These inherent trade-offs motivate the multiobjective optimization framework adopted in this thesis. In particular, the min-max scalarization introduced in the subsequent section treats these metrics jointly, ensuring balanced performance across reconstruction, compression, and perceptual criteria.

3.4 Min–Max Objective as Robust Scalarization

The evaluation metrics introduced in Section 3.3 capture complementary but competing aspects of tensor network (TN) performance. Optimizing any single metric in isolation can lead to undesirable solutions, such as highly accurate but impractically large models or strongly compressed representations with poor perceptual quality. This section introduces a robust min–max scalarization framework that enables balanced multiobjective optimization while explicitly enforcing feasibility constraints.

3.4.1 Multiobjective Optimization Perspective

Let $\mathbf{m} = (m_1, m_2, \dots, m_K)$ denote a vector of performance metrics, where each component corresponds to a distinct objective. In this work, the metrics include relative squared error (RSE), compression efficiency, PSNR, and SSIM. The resulting optimization problem is inherently multiobjective and does not admit a unique scalar optimum without additional structure.

A common approach is to form a weighted sum of objectives; however, weighted-sum methods are sensitive to scale differences and can obscure trade-offs. An alternative, well established in robust and minimax optimization, is to minimize the maximum normalized deviation across objectives. This formulation seeks solutions that are uniformly good rather than optimal for a single criterion [29, 30].

3.4.2 Reference-Normalized Deviations

To make different metrics comparable, each objective is normalized relative to a reference value. Let $(R_{\text{ref}}, L_{\text{ref}}, P_{\text{ref}}, S_{\text{ref}})$ denote baseline values for RSE, log-compression ratio, PSNR, and SSIM, respectively. In this work, the baseline corresponds to a canonical chain tensor network evaluated under identical training conditions.

For a candidate topology $\mathcal{G}$, the normalized deviations are defined as

$$\begin{aligned} b_{\text{RSE}}(\mathcal{G}) &= \frac{\text{RSE}(\mathcal{G})}{R_{\text{ref}}}, \\ b_{\text{CR}}(\mathcal{G}) &= \max\left(0, \frac{L_{\text{ref}} - \log \rho(\mathcal{G})}{L_{\text{ref}}}\right), \\ b_{\text{PSNR}}(\mathcal{G}) &= \max\left(0, \frac{P_{\text{ref}} - \text{PSNR}(\mathcal{G})}{P_{\text{ref}}}\right), \\ b_{\text{SSIM}}(\mathcal{G}) &= \max\left(0, \frac{S_{\text{ref}} - \text{SSIM}(\mathcal{G})}{S_{\text{ref}}}\right), \end{aligned}$$

where ρ denotes the compression ratio. Each term measures the relative shortfall of the candidate model with respect to the baseline; values equal to zero indicate no degradation.

3.4.3 Min–Max Scalarization

The scalar objective minimized in this thesis is defined as

$$J_4(\mathcal{G}) = \max(w_{\text{RSE}} b_{\text{RSE}}, w_{\text{CR}} b_{\text{CR}}, w_{\text{PSNR}} b_{\text{PSNR}}, w_{\text{SSIM}} b_{\text{SSIM}}),$$

where $w. > 0$ are user-specified weights controlling the relative importance of each objective. This formulation corresponds to a weighted ℓ_∞ norm in the space of normalized deviations and is a standard construction in minimax and robust optimization [29, 30].

Minimizing J_4 seeks tensor network structures whose worst relative degradation across all metrics is as small as possible. As a result, no single objective can dominate the optimization, and improvements in one metric cannot compensate for severe degradation in another.

3.4.4 Feasibility-Aware Evaluation

Crucially, the min–max objective is evaluated only for *feasible* tensor network structures. Prior to training and metric evaluation, candidate topologies are screened using structural constraints on node degree, number of edges, parameter count, and estimated intermediate contraction size. Candidates violating these constraints are discarded without further evaluation.

This early feasibility filtering aligns with theoretical insights on contraction complexity and prevents catastrophic cost explosions during optimization. As a consequence, the min–max objective operates over a restricted but tractable search space, ensuring both robustness and computational safety.

3.4.5 Interpretation and Design Rationale

From a robust optimization viewpoint, the proposed scalarization can be interpreted as seeking solutions that perform well under the worst-case objective among a predefined set of criteria. This is particularly appropriate in tensor network struc-

ture search, where different applications may prioritize accuracy, compression, or perceptual quality differently.

By grounding the objective in minimax principles, the proposed framework avoids ad hoc trade-off tuning and provides a principled mechanism for balanced tensor network design. The resulting optimization problem forms the basis for the neighborhood-based structure search algorithm described in the next section.

3.5 Topology Search Algorithm

This section presents the proposed topology search procedure for arbitrary-graph tensor networks, together with the robust min-max objective introduced in Section 3.4. The algorithm performs a constrained neighborhood exploration in the discrete space of graph structures (bit-encoded edges), evaluates candidates using a train-evaluate loop, and selects the topology that minimizes the worst normalized degradation across the metrics (RSE, compression, PSNR, SSIM). The implementation follows an “evaluate-then-move” local search strategy with early feasibility screening to avoid memory/time explosions during contraction and training .

3.5.1 Search Space and Neighborhood Definition

We consider an undirected graph over N nodes (in our experiments $N = 8$), encoded by a binary vector $\mathbf{b} \in \{0, 1\}^{N(N-1)/2}$ that indicates which of the possible edges (i, j) is present . A *neighbor* of $\mathbf{b}$ is obtained by flipping a small number of bits; in the default configuration, the algorithm samples k random bit positions and generates the corresponding k one-bit-flip neighbors . This yields a lightweight neighborhood exploration step that is compatible with expensive evaluation (training + contraction).

3.5.2 Feasibility Screening (Safety Constraints)

Before any expensive evaluation, each candidate topology is screened using explicit constraints on: maximum node degree, maximum number of edges, estimated parameter count, and a rough upper bound on intermediate contraction size . Infeasible candidates are discarded immediately (no training), which substantially reduces wasted computation and prevents out-of-memory failures during contraction/training.

3.5.3 Candidate Evaluation: Train–Reconstruct–Score

For a feasible candidate topology $\mathbf{b}$ and fixed bond dimension q (rank), we instantiate the arbitrary-graph TN model by constructing tensor cores whose shapes are determined by node degrees, then building a single einsum equation from the incident edge labels and physical labels. The model is trained to minimize relative squared reconstruction loss via Adam (optionally followed by LBFGS refinement). After training, we compute: (i) tensor-domain reconstruction error (RSE), (ii) compression ratio ρ via parameter count, (iii) PSNR and SSIM on the reconstructed image.

3.5.4 How the Min–Max Objective Operates During Search

Let $\text{RSE}(\mathbf{b})$, $\log \rho(\mathbf{b})$, $\text{PSNR}(\mathbf{b})$, and $\text{SSIM}(\mathbf{b})$ denote the metrics for topology $\mathbf{b}$. The algorithm first evaluates a *baseline* topology (default: chain/TT-like) and uses it to define reference values $(R_{\text{ref}}, L_{\text{ref}}, P_{\text{ref}}, S_{\text{ref}})$. Each candidate is then scored by the robust scalarization

$$J_4(\mathbf{b}) = \max\left(w_{\text{RSE}} b_{\text{RSE}}(\mathbf{b}), w_{\text{CR}} b_{\text{CR}}(\mathbf{b}), w_{\text{PSNR}} b_{\text{PSNR}}(\mathbf{b}), w_{\text{SSIM}} b_{\text{SSIM}}(\mathbf{b})\right),$$

where $b_{\cdot}$ are reference-normalized shortfalls (clipped at 0) and $w_{\cdot} > 0$ are user-specified weights. Minimizing J_4 favors topologies whose *worst* (most degraded) metric is as small as possible: improving one metric cannot compensate for severely damaging another.

3.5.5 Pseudo-code

The flowchart in Fig. 3.1 illustrates the control flow of Algorithm 1. Starting from a baseline tensor network topology, the algorithm iteratively explores a local neighborhood of candidate structures. Infeasible candidates are discarded early to prevent excessive computational cost, while feasible ones are trained and evaluated using the robust min–max objective J_4 , which aggregates reconstruction error, compression efficiency, and perceptual quality metrics. The current topology is updated only when an improvement in J_4 is achieved, and the search terminates when no better topology is found within the neighborhood.

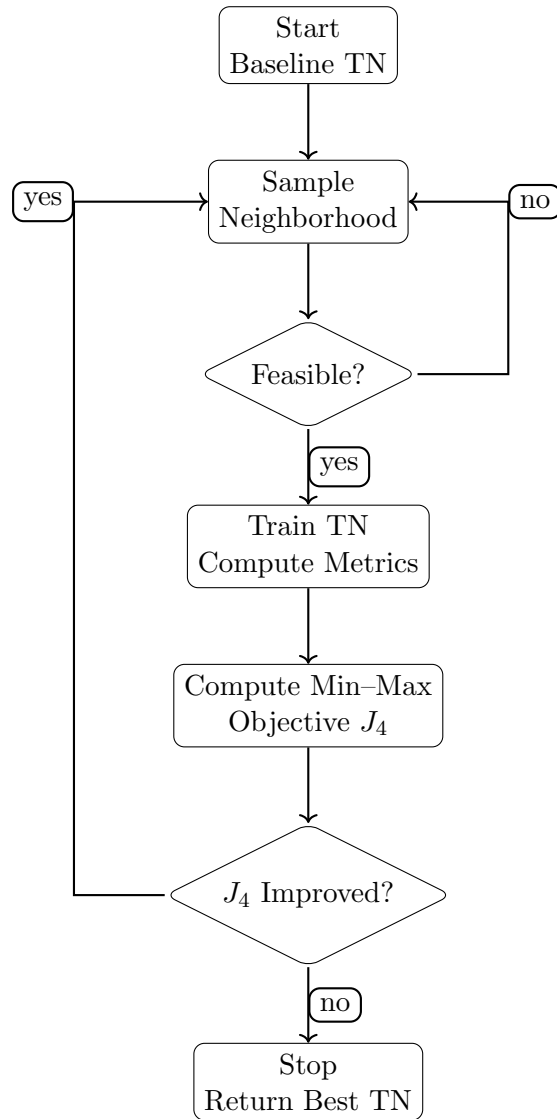


Figure 3.1: Flowchart of the min-max topology search algorithm.

Algorithm 1: Min–Max Topology Search via Neighborhood Exploration

Input: Image tensor $\mathcal{X}$, rank q , search budget T , neighbors k

Input: Safety constraints $\mathcal{S}$, weights $\mathbf{w}$

Output: Best topology $\mathbf{b}^*$

Initialize baseline topology $\mathbf{b}_0$ (chain/TT);

Evaluate baseline and compute reference metrics;

Set $\mathbf{b} \leftarrow \mathbf{b}_0$, $\mathbf{b}^* \leftarrow \mathbf{b}_0$;

for $t = 1$ **to** T **do**

 Sample k one-bit neighbors of $\mathbf{b}$;

$J_{\min} \leftarrow +\infty$;

foreach *neighbor* $\mathbf{b}'$ **do**

if $\mathbf{b}'$ *violates* $\mathcal{S}$ **then**

 continue;

 Train TN with topology $\mathbf{b}'$;

 Compute RSE, compression, PSNR, SSIM;

 Compute $J_4(\mathbf{b}') = \max_i w_i b_i$;

if $J_4(\mathbf{b}') < J_{\min}$ **then**

$J_{\min} \leftarrow J_4(\mathbf{b}')$;

$\mathbf{b}_{\text{best}} \leftarrow \mathbf{b}'$;

if $J_{\min} < J_4(\mathbf{b})$ **then**

$\mathbf{b} \leftarrow \mathbf{b}_{\text{best}}$;

if $J_{\min} < J_4(\mathbf{b}^*)$ **then**

$\mathbf{b}^* \leftarrow \mathbf{b}$;

else

 break;

Retrain final model using $\mathbf{b}^*$;

return $\mathbf{b}^*$;

3.5.6 Remarks on Practical Configuration

In the default configuration used in our experiments, the search budget is deliberately small (a few steps and a few neighbor evaluations per step), reflecting the fact that each evaluation includes model training and image-metric computation. The algorithm therefore behaves as a cost-aware greedy local search, where feasibility guards and the min–max criterion jointly ensure that accepted moves are (i) tractable

and (ii) balanced across reconstruction fidelity, compression, and perceptual quality. Finally, after a topology is selected, it is re-trained with an increased optimization budget to obtain the final reported metrics, ensuring that topology comparisons are not unduly biased by short training runs during search .

3.6 Topology Search via Feasibility-First Local Exploration

This section presents the proposed topology search algorithm, which constitutes the main methodological contribution of this thesis. The algorithm is designed as a lightweight, feasibility-aware alternative to existing tensor network (TN) structure learning methods. It combines local neighborhood exploration with explicit feasibility screening and a robust min–max acceptance criterion, enabling practical structure adaptation under tight computational constraints.

3.6.1 Design Principles

The proposed method is guided by three core principles.

First, topology exploration is performed locally rather than globally. Instead of searching over unrestricted graph spaces, the algorithm explores a small neighborhood of candidate topologies around a current solution. This choice reflects the high cost of evaluating TN structures and aligns with the observation that meaningful improvements often arise from small structural modifications.

Second, feasibility constraints are enforced *prior* to any expensive evaluation. Candidate topologies are screened using structural and cost-based criteria (e.g., node degree, parameter count, contraction size bounds), ensuring that only tractable architectures are trained and evaluated.

Third, acceptance decisions are based on a robust min–max objective rather than single-metric improvement. This prevents the search from over-optimizing one criterion at the expense of others and promotes balanced performance across reconstruction accuracy, compression, and perceptual quality.

3.6.2 Local Neighborhood Operator

Topology is encoded as a binary vector indicating the presence or absence of edges in an undirected TN graph. The neighborhood of a topology is defined via single-bit flips, corresponding to the addition or removal of individual edges. At each iteration, a small subset of such neighbors is sampled uniformly at random, yielding a candidate set whose size is independent of the total topology space.

This local operator enables efficient exploration while maintaining interpretability: each move corresponds to a minimal and explicit change in network connectivity. Unlike evolutionary approaches, no crossover or population management is required.

3.6.3 Search Procedure and Acceptance Criterion

For each sampled neighbor, feasibility checks are applied immediately. Only feasible candidates proceed to training and evaluation. Each feasible topology is trained under a fixed budget and scored using the min–max objective introduced in Section 3.4.

The search follows a greedy accept-if-improves strategy. If the best candidate in the neighborhood yields a strict improvement in the min–max objective, the current topology is updated. Otherwise, the search terminates early. This stopping rule prevents unnecessary evaluations once local optimality with respect to the chosen neighborhood has been reached.

3.6.4 Relation to Existing Structure Search Methods

The proposed algorithm is conceptually related to Tensor Network Architecture Learning (TnALE) [21], which also alternates between structural updates and parameter optimization. However, TnALE allows a broader class of structural moves and typically incurs higher computational cost due to repeated re-optimization. In contrast, the method proposed here restricts exploration to a lightweight local neighborhood and emphasizes early feasibility pruning.

Compared to permutation-based local search [20], the proposed approach is not limited to fixed topological families and can introduce or remove edges, enabling a richer class of architectures. At the same time, it avoids the large population-based overhead of evolutionary search methods and serves as a practical middle ground between rigid local optimization and expensive global topology search.

Overall, the proposed feasibility-first local exploration strategy provides a simple,

scalable, and effective mechanism for tensor network topology adaptation. By combining local structural moves, early pruning, and robust acceptance criteria, the algorithm achieves meaningful structure learning while remaining compatible with realistic computational budgets.

3.7 Training Routine: Adam to L-BFGS with Gradient Clipping

Training tensor network (TN) models with arbitrary graph structures presents non-trivial optimization challenges. The loss landscape is typically nonconvex, gradients may vary significantly in magnitude across tensor cores, and intermediate contractions can amplify instabilities. To address these issues, this thesis adopts a two-stage training routine combining adaptive first-order optimization with optional second-order refinement, together with explicit gradient clipping.

3.7.1 Stage I: Adaptive Optimization with Adam

The primary training phase uses the Adam optimizer, introduced by Kingma and Ba [31]. Adam adapts learning rates on a per-parameter basis using estimates of first and second moments of the gradients, making it well suited for high-dimensional and ill-conditioned optimization problems.

In the context of tensor networks, Adam provides several practical advantages. First, it allows rapid initial convergence from random initialization, which is particularly important when evaluating many candidate topologies during structure search. Second, its adaptive step sizes help mitigate imbalance between gradients associated with different tensor cores or bond dimensions.

During topology search, all candidate models are trained using Adam under a fixed and relatively small budget. This ensures fair comparison between different structures while keeping evaluation costs manageable.

3.7.2 Stage II: Optional Refinement with L-BFGS

After a topology has been selected, an optional second optimization stage using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm is applied. L-BFGS is a quasi-Newton method that approximates second-order curvature

information using a limited history of gradient updates [32].

While more expensive per iteration than Adam, L-BFGS often achieves superior final convergence for smooth reconstruction objectives. In this work, it is used only for final model refinement and not during topology search, where its computational cost would be prohibitive. This separation allows the method to benefit from the fast exploration capabilities of Adam and the fine-grained convergence of second-order methods.

3.7.3 Gradient Clipping for Stability

To further improve training stability, gradient clipping is applied during both optimization stages. Specifically, gradients are clipped to a maximum norm before parameter updates.

Gradient clipping was originally proposed as a mechanism to prevent exploding gradients in deep and recurrent neural networks [33]. In tensor network training, similar issues can arise due to large intermediate contractions or highly connected graph structures. Clipping ensures that no single update step destabilizes the optimization process, particularly during early training or when exploring new topologies.

3.7.4 Practical Considerations

The combination of Adam, optional L-BFGS refinement, and gradient clipping provides a robust and flexible training routine. Adam enables efficient evaluation during structure search, while L-BFGS improves final reconstruction quality for the selected topology. Gradient clipping acts as a safeguard against numerical instability and improves reproducibility across different runs.

Overall, this training strategy balances optimization robustness with computational efficiency and is well suited to the iterative topology search framework proposed in this thesis.

Chapter 4

Experimental results

4.1 Experimental Setup

The experiments in this chapter are conducted using a subset of the **BSR BSDS500 dataset**, which is widely used for benchmarking image processing and reconstruction algorithms. From this dataset, we select **10 images** for evaluation (see Fig. 4.1). Each image is converted to **grayscale** and resized to a resolution of 256×256 **pixels**. These images are then tensorized and used as input for the tensor network reconstruction experiments.

Since each grayscale image has a resolution of 256×256 pixels, each original image contains 65,536 parameters. For this reason, the number of parameters used by each tensor network model is an important factor when evaluating reconstruction performance. A model with a smaller number of parameters typically achieves a higher compression ratio but may suffer from reduced reconstruction quality due to limited representational capacity. Conversely, a model with a large number of parameters can produce higher-quality reconstructions, but at the cost of increased computational complexity and lower compression efficiency. Therefore, an effective tensor network model should balance these two competing aspects, achieving acceptable reconstruction quality while maintaining a compact representation.

To evaluate the performance of different topology search strategies, several quantitative metrics are used, including Mean Squared Error (MSE), Relative Squared Error (RSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and the logarithmic compression ratio (LogCR).

Lower values of **MSE** and **RSE** indicate better reconstruction accuracy, as they measure the difference between the reconstructed image and the original one. Conversely,



Figure 4.1: The 10 grayscale images selected from the BSR BSDS500 dataset used in the experiments. All images are resized to 256×256 pixels.

higher values of **PSNR** and **SSIM** correspond to better perceptual reconstruction quality and stronger structural similarity between the reconstructed and reference images.

For the compression metric, a higher value of **LogCR** indicates a greater compression ratio, meaning that fewer parameters are required to represent the image. However, increasing compression may negatively affect reconstruction quality. Therefore, the goal is to achieve a balanced trade-off between reconstruction accuracy and compression efficiency, where the model maintains high PSNR and SSIM while keeping MSE and RSE low and achieving a sufficiently high compression ratio.

Furthermore, to analyze the influence of model capacity on reconstruction performance, the experiments are conducted across several tensor ranks. In this study, **rank 4** is considered a *low-rank* setting, representing models with limited representational capacity. **Rank 6** is treated as a *medium-rank* configuration that provides a balance between expressivity and computational cost. Finally, **rank 8** is considered a *high-rank* setting, where the tensor network has significantly greater representational flexibility. The experimental results presented in the following sections analyze and compare the performance of different topology search strategies under these three rank regimes.

4.2 Results for Rank 4

Figure 4.2 presents the experimental results for tensor rank 4. At this relatively low rank, the model capacity is limited and topology plays a significant role in determining reconstruction quality.

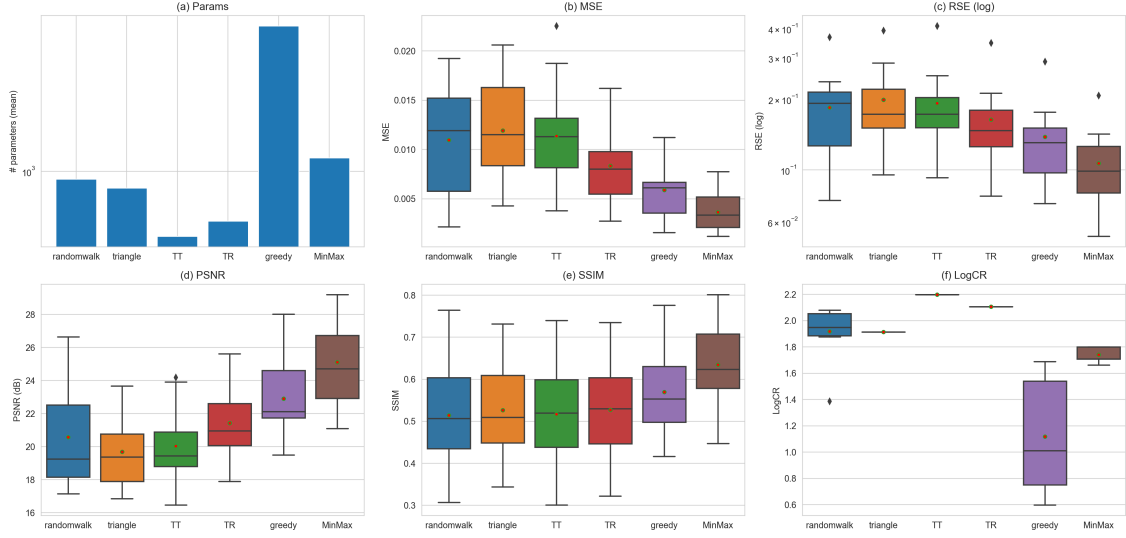


Figure 4.2: Experimental results for tensor rank 4. The plots show parameter count, MSE, RSE, PSNR, SSIM, and logarithmic compression ratio (LogCR) for different topology search strategies.

The parameter comparison in Fig. 4.2(a) shows that TT and TR have the smallest parameter counts, while the greedy method introduces substantially larger models. The proposed MinMax approach maintains a moderate parameter count while avoiding excessive model complexity.

In terms of reconstruction error, both MSE and RSE plots in Fig. 4.2(b)–(c) demonstrate that the MinMax method achieves the lowest error among all tested methods. Baseline approaches such as Random Walk and Triangle produce significantly larger errors due to their less structured topology exploration.

The perceptual metrics further support this observation. As shown in Fig. 4.2(d) and Fig. 4.2(e), the MinMax approach achieves the highest PSNR and SSIM values, indicating improved reconstruction fidelity and structural similarity with the original images. Although the greedy strategy also performs relatively well, its performance is less consistent across runs.

Finally, the compression metric shown in Fig. 4.2(f) indicates that the MinMax approach achieves competitive compression while maintaining superior reconstruction quality.

Table 4.1 reports the average values of the evaluation metrics across 10 grayscale images for tensor rank 4. As shown, the proposed MinMax method achieves the lowest reconstruction errors (MSE and RSE) and the highest perceptual quality metrics (PSNR and SSIM), while maintaining a competitive compression ratio compared

Table 4.1: Average performance metrics for tensor rank 4 across 10 images from the BSR BSDS500 dataset.

Model	n_params	MSE	RSE	PSNR	SSIM	LogCR
randomwalk	902	0.0109	0.1847	20.5782	0.5141	1.9180
triangle	800	0.0119	0.1990	19.6873	0.5263	1.9133
TT	416	0.0113	0.1926	20.0372	0.5167	2.1973
TR	512	0.0083	0.1641	21.4380	0.5270	2.1072
greedy	7106	0.0059	0.1383	22.9124	0.5694	1.1174
MinMax	1198	0.0036	0.1069	25.1196	0.6343	1.7407

with the baseline methods.

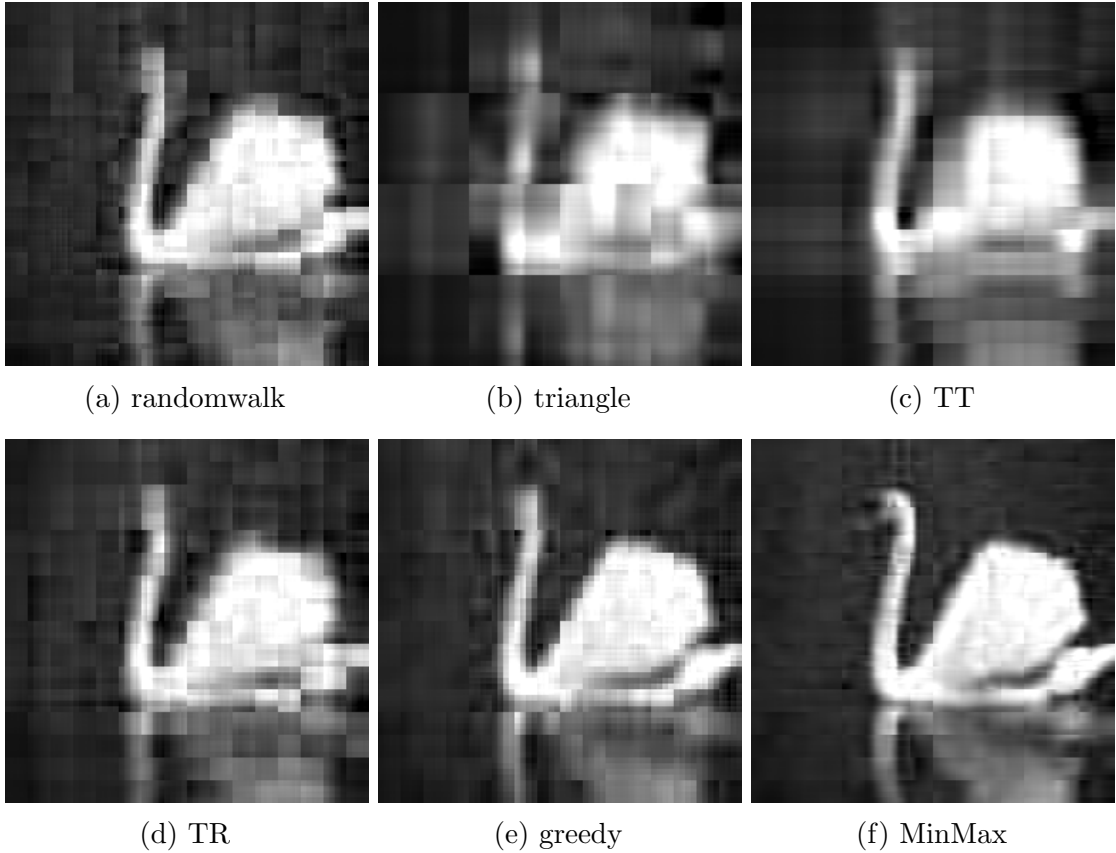


Figure 4.3: Rank-4 reconstruction examples for six topology strategies: (a) randomwalk, (b) triangle, (c) TT, (d) TR, (e) greedy, and (f) MinMax.

Figure 4.3 provides qualitative reconstruction examples for the six evaluated models at rank 4. Since rank 4 corresponds to a low-rank configuration, the tensor networks have limited representational capacity, and high-frequency details and fine structures cannot be fully captured. As a result, all methods exhibit some degree of smoothing and loss of texture, which is expected in a low-rank regime.

Among the baselines, randomwalk (Fig. 4.3a) and triangle (Fig. 4.3b) tend to produce

noisier or less structured reconstructions. TT (Fig. 4.3c) often yields overly smooth outputs due to its restrictive chain topology, while TR (Fig. 4.3d) typically improves visual quality by allowing cyclic correlations. The greedy method (Fig. 4.3e) often achieves stronger reconstructions, but at the cost of substantially increased model size.

The proposed MinMax method (Fig. 4.3f) provides the most visually consistent reconstruction in this low-rank setting, preserving more structural content while avoiding excessive artifacts. Although rank 4 cannot achieve high-fidelity reconstructions in general, the MinMax results remain qualitatively good and acceptable compared with the competing methods, supporting the quantitative improvements reported in the error and perceptual metrics.

4.3 Results for Rank 6

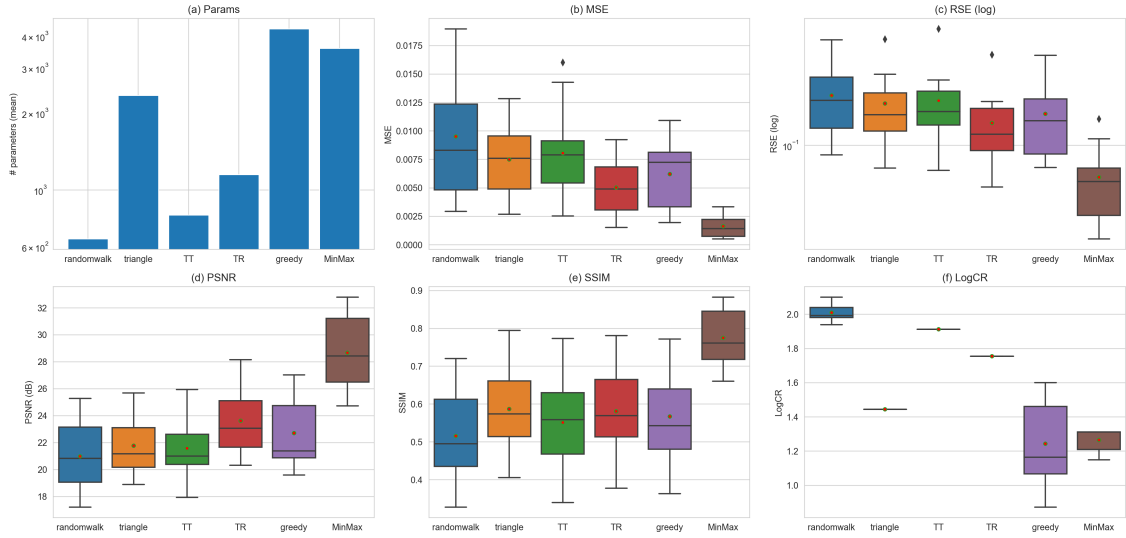


Figure 4.4: Experimental comparison of tensor network topologies for rank 6.

The results for tensor rank 6 are illustrated in Fig. 4.4. With increased rank capacity, the gap between different topology strategies becomes more apparent.

The parameter comparison in Fig. 4.4(a) indicates that the greedy approach introduces the largest model size, while TT remains compact but less expressive.

The error metrics in Fig. 4.4(b)–(c) confirm that the MinMax method continues to achieve the lowest reconstruction error. The difference between MinMax and the other approaches becomes more pronounced as rank increases.

Perceptual evaluation using PSNR and SSIM in Fig. 4.4(d)–(e) again demonstrates the superiority of the MinMax method. The improvement in SSIM is particularly important, as it indicates better preservation of structural image information.

Compression results in Fig. 4.4(f) show that although MinMax uses slightly more parameters than TT, it still achieves strong compression compared with greedy topology search.

Table 4.2: Average performance metrics for tensor rank 6 across 10 images from the BSR BSDS500 dataset.

Model	n_params	MSE	RSE	PSNR	SSIM	LogCR
randomwalk	644	0.0095	0.1704	20.9886	0.5158	2.0103
triangle	2352	0.0075	0.1566	21.7781	0.5873	1.4450
TT	800	0.0080	0.1615	21.5886	0.5518	1.9133
TR	1152	0.0050	0.1270	23.6374	0.5813	1.7550
greedy	4278	0.0062	0.1403	22.7179	0.5678	1.2445
MinMax	3588	0.0016	0.0712	28.6794	0.7753	1.2658

Table 4.2 reports the average values of the evaluation metrics across 10 grayscale images for tensor rank 6. The results show that the proposed MinMax approach achieves the lowest reconstruction errors (MSE and RSE) and the highest perceptual quality metrics (PSNR and SSIM), indicating superior reconstruction performance compared with the baseline methods.

Figure 4.5 shows qualitative reconstruction examples obtained with the six evaluated topology strategies at rank 6. Although rank 6 corresponds to a medium-rank configuration, it still imposes a limited representational budget compared with higher-rank models, and some loss of fine texture and high-frequency details is expected. In particular, subtle edges and small structures may appear smoother than in the original image, since the tensor network cannot allocate sufficient capacity to all local variations.

Among the baselines, randomwalk (Fig. 4.5a) and triangle (Fig. 4.5b) tend to produce reconstructions with weaker structural consistency. TT (Fig. 4.5c) often exhibits additional smoothing due to its restrictive chain-like connectivity, while TR (Fig. 4.5d) typically improves visual quality by capturing richer correlations through cyclic connections. The greedy method (Fig. 4.5e) can also improve reconstruction quality, but it may do so by increasing model complexity.

The proposed MinMax approach (Fig. 4.5f) provides the most visually consistent reconstruction among the tested methods. It preserves more structural detail while avoiding strong artifacts, yielding a reconstruction that is qualitatively good and

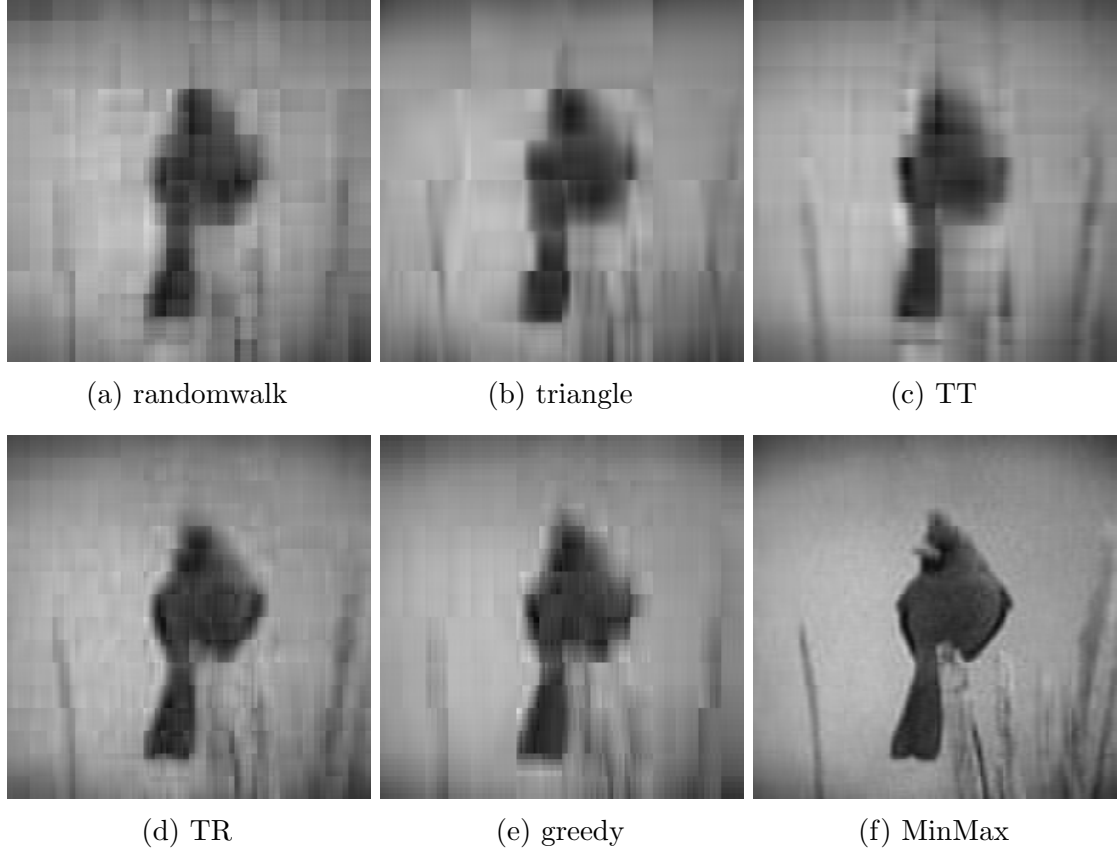


Figure 4.5: Rank-6 reconstruction examples for six topology strategies: (a) randomwalk, (b) triangle, (c) TT, (d) TR, (e) greedy, and (f) MinMax.

acceptable within the rank-6 capacity regime. This observation is consistent with the quantitative results, where MinMax achieves the best trade-off between reconstruction error and perceptual quality.

4.4 Results for Rank 8

The highest-rank experiments are shown in Fig. 4.6. As expected, all methods benefit from increased model capacity, resulting in improved reconstruction metrics.

However, the MinMax method still provides the best overall performance. It consistently achieves the lowest MSE and RSE, as shown in Fig. 4.6(b)–(c), indicating more accurate reconstruction.

The perceptual metrics presented in Fig. 4.6(d)–(e) show the largest PSNR and SSIM values for MinMax, confirming that the proposed topology search approach produces the highest quality reconstructions.

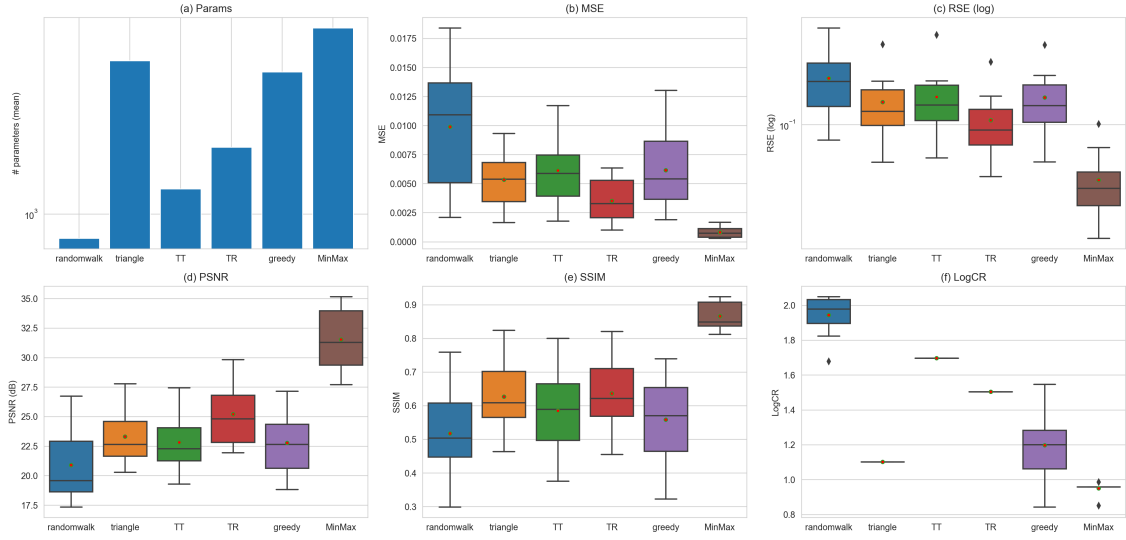


Figure 4.6: Experimental results for tensor rank 8. The MinMax approach consistently achieves the best reconstruction performance across the evaluated metrics.

The compression analysis in Fig. 4.6(f) indicates that although some baseline methods achieve slightly smaller parameter counts, they do so at the cost of reduced reconstruction quality.

Table 4.3: Average performance metrics for tensor rank 8 across 10 images from the BSR BSDS500 dataset.

Model	n_params	MSE	RSE	PSNR	SSIM	LogCR
randomwalk	772	0.0099	0.1753	20.8992	0.5182	1.9449
triangle	5184	0.0053	0.1314	23.3281	0.6271	1.1018
TT	1312	0.0061	0.1401	22.8253	0.5855	1.6985
TR	2048	0.0035	0.1057	25.2274	0.6363	1.5051
greedy	4603	0.0061	0.1389	22.7830	0.5588	1.1987
MinMax	7356	0.0008	0.0514	31.5378	0.8665	0.9512

Table 4.3 reports the average values of the evaluation metrics across 10 grayscale images for tensor rank 8. As shown, the proposed MinMax method achieves the lowest reconstruction errors (MSE and RSE) and the highest perceptual quality metrics (PSNR and SSIM), demonstrating superior reconstruction performance compared with the baseline methods while maintaining competitive compression.

Figure 4.7 presents qualitative reconstruction examples for the six evaluated methods at rank 8. Since rank 8 corresponds to a high-rank configuration, the tensor networks have substantially greater representational capacity than in the low- and medium-rank settings. As a result, reconstructions generally preserve sharper edges and finer structures, and visual artifacts are reduced compared with rank 4 and rank 6.

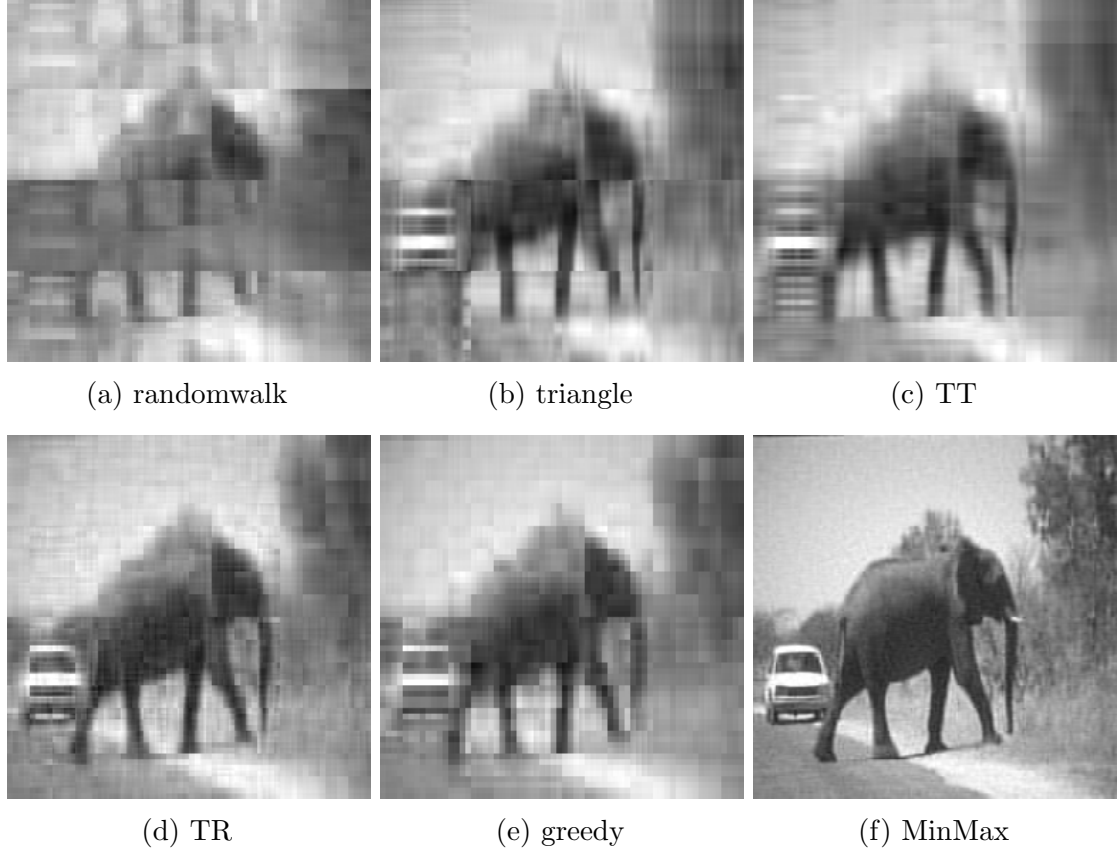


Figure 4.7: Rank-8 reconstruction examples for six topology strategies: (a) randomwalk, (b) triangle, (c) TT, (d) TR, (e) greedy, and (f) MinMax.

Nevertheless, some small-scale texture loss may still occur depending on the topology and the compression–quality trade-off.

Among the baselines, randomwalk (Fig. 4.7a) tends to produce reconstructions with weaker structure, while triangle (Fig. 4.7b) improves structural consistency but can still miss fine details. TT (Fig. 4.7c) often appears smoother due to its restrictive chain connectivity, whereas TR (Fig. 4.7d) typically captures richer correlations and yields clearer structure. The greedy method (Fig. 4.7e) may improve reconstruction in some cases, but it can be less stable and may increase complexity unnecessarily.

The proposed MinMax approach (Fig. 4.7f) provides the most visually faithful reconstruction among the tested methods. It preserves structural content and contrast more effectively while avoiding noticeable artifacts, yielding a reconstruction that is qualitatively strong and acceptable. This qualitative observation is consistent with the quantitative results, where MinMax achieves the lowest reconstruction errors and the highest PSNR and SSIM values at rank 8.

4.5 Topology Adaptation in Tensor Networks

One of the central challenges in tensor network (TN) modeling is the selection of an appropriate network topology. Traditional tensor network approaches typically rely on fixed architectures such as chain structures (Tensor Train), tree structures (Hierarchical Tucker), or cyclic structures (Tensor Ring). Although these predefined topologies are computationally convenient, they implicitly impose structural assumptions on the relationships between tensor modes. In practice, however, real-world data such as natural images often contain complex correlations that cannot be adequately captured by a single predefined topology. Consequently, selecting an appropriate network structure becomes an important problem when designing efficient tensor network models.

Figure 4.8 illustrates several examples of tensor network topologies discovered during the reconstruction process. Each image in the top row corresponds to an example image from the dataset, while the bottom row shows the corresponding tensor network topology learned for that image. These examples highlight an important observation: different images tend to favor different connectivity structures. For instance, images containing strong directional patterns or structured textures may benefit from topologies that emphasize certain pairwise interactions between tensor modes. Conversely, simpler scenes may require fewer connections and therefore admit more compact representations.

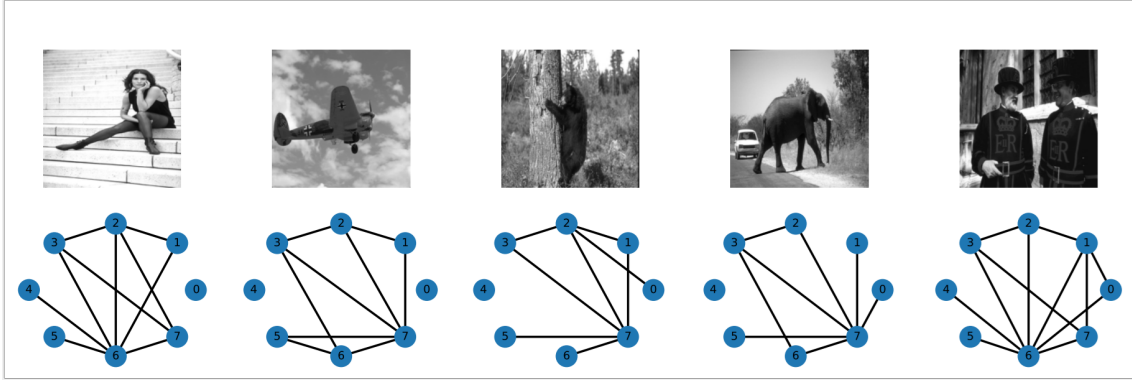


Figure 4.8: Examples of tensor network topologies discovered for different images. The first row shows sample images from the dataset, while the second row illustrates the corresponding tensor network structures identified during the topology search process.

The diversity of the learned structures shown in Fig. 4.8 demonstrates that a single fixed topology cannot optimally represent all images. When a topology is fixed in advance, the model may fail to capture important correlations between

tensor dimensions. This limitation may lead to suboptimal reconstructions, higher reconstruction error, or inefficient parameter utilization. For example, a chain topology such as the Tensor Train restricts interactions to neighboring nodes, while tree-based structures impose hierarchical dependencies that may not align with the intrinsic relationships present in the data.

To address this limitation, topology search methods aim to discover a more suitable network structure directly from the data. Instead of committing to a predefined architecture, the search process explores a set of candidate connectivity patterns and selects the one that achieves the best balance between reconstruction quality, model complexity, and compression efficiency. This adaptive strategy allows the tensor network to better align with the underlying structure of the data, leading to improved performance in image reconstruction tasks.

The examples in Fig. 4.8 clearly illustrate that the optimal topology varies depending on the image content. Some images favor sparse connectivity patterns, while others benefit from richer interaction structures between tensor modes. Therefore, topology search plays a crucial role in enabling tensor networks to adapt to different data characteristics and achieve higher reconstruction quality without unnecessarily increasing model complexity.

4.6 Overall Discussion

Across all tested tensor ranks, the experimental results demonstrate a consistent trend. The topology selected by the MinMax optimization framework provides the best balance between reconstruction accuracy, perceptual quality, and compression efficiency.

While baseline methods such as TT and TR offer compact parameter representations, their fixed topology limits their expressive power. Random exploration strategies, such as Random Walk, show high variance and unreliable performance.

Greedy topology search improves reconstruction quality but tends to produce excessively large models. In contrast, the proposed MinMax approach explicitly balances multiple objectives, allowing it to select structures that achieve both strong reconstruction accuracy and efficient parameter usage.

In summary, the experimental evaluation demonstrates that topology selection plays a critical role in tensor network performance. Across all tested ranks, the MinMax topology search method consistently achieves the best reconstruction quality, the

highest perceptual metrics, and competitive compression ratios.

These results confirm that the proposed MinMax framework provides a robust and effective strategy for adaptive tensor network topology learning.

Chapter 5

Conclusion

5.1 What Was Learned About Topology as Inductive Bias

A central theme of this thesis has been the role of tensor network (TN) topology as an explicit inductive bias. Rather than viewing topology as a fixed architectural choice inherited from canonical decompositions, this work treated it as a variable to be explored and optimized under computational constraints. The experimental and theoretical findings support several key observations.

5.1.1 Topology Governs Expressivity

The connectivity pattern of a tensor network determines the class of functions or tensors that can be represented efficiently. As shown by Glasser et al. [15], tensor-network factorizations exhibit different expressive capacities depending on their graph structure. Certain probability distributions or interaction patterns can be compactly represented in one topology but require exponentially larger ranks in another.

Our empirical results align with this theoretical insight. Topologies that better align with spatial structure (e.g., through locality-aware tensorization or selective edge additions) achieved improved reconstruction quality at comparable or lower parameter counts. Conversely, overly sparse or misaligned structures often required inflated ranks to compensate for structural limitations.

These observations reinforce the interpretation of topology as an inductive bias: it encodes assumptions about how variables interact. When the topology reflects the

intrinsic correlation structure of the data, the model achieves higher expressivity per parameter.

5.1.2 Topology Determines Computational Cost

While increased connectivity can enhance expressivity, it also affects computational tractability. The contraction complexity of a tensor network depends critically on graph properties such as treewidth. Markov and Shi [17] established that the cost of contracting a general tensor network scales exponentially with the treewidth of its underlying graph.

This theoretical result was directly reflected in the behavior observed during topology search. Highly connected candidate graphs frequently violated feasibility constraints due to excessive intermediate tensor sizes. Even when such structures improved reconstruction error, they were often computationally impractical.

Thus, topology simultaneously controls representational power and contraction cost. The tension between these two aspects forms the fundamental trade-off addressed in this thesis.

5.1.3 Balanced Structure Through Robust Optimization

The min-max scalarization framework introduced in Chapter 3 provided a practical mechanism for navigating the expressivity-cost trade-off. Rather than optimizing reconstruction error alone, the algorithm favored topologies that performed well across multiple metrics, including compression and perceptual quality.

This balanced objective often led to intermediate structures: richer than simple chains, but significantly sparser than dense graphs. Such architectures achieved strong empirical performance while remaining within computational limits.

5.1.4 Implications for Tensor Network Design

The findings of this thesis suggest that tensor network topology should not be treated as a fixed design choice. Instead, it should be regarded as a controllable inductive bias that can be adapted to the problem at hand. However, this flexibility must be constrained by computational feasibility, as dictated by contraction complexity theory.

In summary, topology acts as the primary structural prior in tensor networks. It governs both expressivity and computational cost, and effective TN design requires balancing these two dimensions. The feasibility-first, locally adaptive search framework developed in this work demonstrates one practical approach to achieving this balance.

5.2 Limitations of the Current Approach

While the proposed feasibility-first local topology search framework demonstrates promising results, several limitations remain. These limitations stem both from computational constraints and from modeling assumptions embedded in the current implementation.

5.2.1 Scaling Beyond Eight Nodes

The present implementation fixes the number of tensor network nodes to $N = 8$, corresponding to the tensorization of 256×256 grayscale images into a 4^8 tensor. Topology is encoded using $N(N - 1)/2 = 28$ binary variables, which already yields 2^{28} possible graph structures in the unrestricted case.

Although the search procedure explores only a small local neighborhood at each step, the combinatorial growth of the topology space becomes prohibitive as N increases. Moreover, contraction complexity grows rapidly with graph connectivity and degree, as reflected in the safety constraints implemented in the search routine. Extending the method to larger node counts would require more sophisticated pruning strategies, hierarchical search, or explicit treewidth control.

5.2.2 Expensive Per-Topology Training

Each candidate topology is evaluated by training a full tensor network model using Adam (and optionally L-BFGS refinement). Even with modest training budgets, this results in substantial computational cost, particularly when multiple neighbors are evaluated per search step.

Unlike surrogate-based or differentiable architecture learning methods, the current approach performs full training for each candidate without parameter reuse across structurally similar topologies. Although this design simplifies implementation and

ensures fair comparison, it limits scalability to larger datasets or deeper search horizons.

5.2.3 Dependence on Thresholds and Weights

The feasibility-first design relies on several user-defined thresholds, including maximum node degree, maximum parameter count, and estimated intermediate contraction size. Similarly, the min-max scalarization depends on weight parameters that balance reconstruction error, compression, and perceptual quality.

While these hyperparameters provide flexibility, they also introduce implicit bias into the search process. Different threshold or weight settings can lead to different selected topologies. Although reasonable default values were chosen and justified experimentally, automatic or adaptive calibration of these parameters remains an open problem.

5.2.4 Contraction Order Not Explicitly Optimized

The current implementation defines tensor contraction through a fixed einsum expression derived directly from the graph structure. While modern frameworks internally apply heuristic contraction ordering, no explicit contraction path optimization is incorporated into the search loop.

As established in contraction complexity theory, contraction order can dramatically affect computational cost. A topology deemed infeasible under naive contraction ordering might become tractable with an optimized contraction path. Conversely, ignoring contraction order may underestimate true runtime for certain graphs.

Integrating contraction path optimization into the search procedure could therefore improve both feasibility assessment and performance estimation.

5.2.5 Local Optimality and Limited Exploration Depth

The topology search strategy is inherently local. At each iteration, only a small subset of one-bit modifications is evaluated, and the search terminates when no immediate improvement in the min-max objective is observed.

While this greedy accept-if-improves strategy ensures efficiency, it may converge to locally optimal topologies that are suboptimal globally. Exploration strategies incor-

porating random restarts, simulated annealing, or hybrid global-local mechanisms could potentially overcome this limitation.

5.2.6 Summary of Limitations

In summary, the proposed approach is computationally practical and conceptually transparent, but its scalability and global optimality are limited by combinatorial growth, per-topology training cost, hyperparameter sensitivity, and contraction-order assumptions. These limitations provide clear directions for future work and motivate the exploration of more scalable, adaptive, and theoretically grounded structure learning strategies.

5.3 Future Directions

The results of this thesis demonstrate that feasibility-first local topology exploration is a viable strategy for adaptive tensor network (TN) structure learning. At the same time, several promising research directions emerge that could extend and strengthen the proposed framework.

5.3.1 Bayesian and Surrogate-Guided Structure Search

One natural extension is the integration of Bayesian or surrogate-based structure search mechanisms. Recent work such as tnGPS [23] and Bayesian Tensor Network Structure Search [22] has shown that Gaussian-process surrogates and uncertainty-aware acquisition functions can dramatically reduce the number of expensive topology evaluations.

Incorporating a surrogate model into the feasibility-first local exploration framework could improve sample efficiency while retaining structural constraints. For example, surrogate predictions could guide which neighbors to evaluate, or approximate the min-max objective prior to full training. Such hybrid approaches would combine the computational safety of local search with the statistical efficiency of Bayesian optimization.

5.3.2 Stronger Integration with Probabilistic Graphical Models

Tensor networks and probabilistic graphical models (PGMs) share a deep structural duality. Robeva and Seigal [16] highlight the equivalence between tensor contraction and marginalization in graphical models, suggesting that structure learning in TNs can be interpreted as graph structure learning in probabilistic inference.

Future work could therefore explore TN topology search in the context of uncertainty-aware inference tasks, such as marginal likelihood estimation, message passing, or UAI-style benchmark problems. This direction would extend the present reconstruction-focused framework toward structured probabilistic modeling and decision-making under uncertainty.

5.3.3 Beyond Reconstruction: Generative and Sequence Modeling

While this thesis focuses primarily on image reconstruction, tensor networks have been successfully applied to probabilistic modeling and sequence learning tasks. For instance, tensor network factorizations have been used in density estimation and generative modeling frameworks [15]. Extending topology search methods to such tasks could reveal different structural preferences and trade-offs.

In sequence modeling, recurrent and autoregressive TN architectures may benefit from adaptive connectivity patterns that capture long-range dependencies efficiently. Investigating topology learning in these settings would broaden the applicability of the proposed framework beyond static reconstruction.

5.3.4 Scaling and Hierarchical Structure Learning

Scaling topology search to larger tensors and higher node counts remains a significant challenge. Hierarchical or multiscale structure search strategies could help manage combinatorial growth. For example, coarse-grained graph search could identify promising substructures, which are then refined locally.

Such hierarchical approaches would align naturally with multiscale tensor network constructions and renormalization-inspired architectures [9]. Combining hierarchical modeling with feasibility-aware pruning may enable structure learning at substantially larger scales.

5.3.5 Tensor Networks in Modern AI Systems

Finally, tensor networks are increasingly positioned as structured, interpretable alternatives or complements to deep neural networks. As discussed by Orús [9], TNs provide compact representations, well-defined contraction semantics, and explicit structural control.

Future work could explore hybrid architectures in which tensor network modules are embedded within modern AI pipelines, leveraging topology adaptation to balance interpretability, efficiency, and expressive power. Such integration would position topology-aware TN learning within the broader landscape of contemporary machine learning research.

5.3.6 Summary

In summary, future research may extend the present framework in three principal directions: improving search efficiency through Bayesian guidance, deepening integration with probabilistic inference, and broadening the scope of applications beyond reconstruction. Together, these directions highlight the continuing evolution of tensor networks from fixed decompositions toward adaptive, data-driven structural models.

Bibliography

- [1] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*. Springer, 2012.
- [3] L. Grasedyck, “Hierarchical singular value decomposition of tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2029–2054, 2010.
- [4] I. V. Oseledets, “Tensor-train decomposition,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [5] S. R. White, “Density matrix formulation for quantum renormalization groups,” *Physical Review Letters*, vol. 69, no. 19, pp. 2863–2866, 1992.
- [6] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Annals of Physics*, vol. 326, no. 1, pp. 96–192, 2011.
- [7] J. I. Cirac, D. Perez-Garcia, N. Schuch, and F. Verstraete, “Matrix product states and projected entangled pair states: Concepts, symmetries, and theorems,” *Reviews of Modern Physics*, vol. 93, no. 4, p. 045003, 2021.
- [8] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Annals of Physics*, vol. 349, pp. 117–158, 2014.
- [9] R. Orús, “Tensor networks for complex quantum systems,” *Nature Reviews Physics*, vol. 1, pp. 538–550, 2019.
- [10] F. Verstraete and J. I. Cirac, “Renormalization algorithms for quantum-many body systems in two and higher dimensions,” *arXiv preprint*, 2004.
- [11] G. Vidal, “Entanglement renormalization,” *Physical Review Letters*, vol. 99, no. 22, p. 220405, 2007.

- [12] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, “Tensor ring decomposition,” *arXiv preprint*, 2016.
- [13] E. Stoudenmire and D. J. Schwab, “Supervised learning with tensor networks,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [14] A. Novikov, D. Podoprikin, A. Osokin, and D. P. Vetrov, “Tensorizing neural networks,” *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [15] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, “Expressive power of tensor-network factorizations for probabilistic modeling,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [16] E. Robeva and J. Seigal, “Duality of graphical models and tensor networks,” *Information and Inference: A Journal of the IMA*, vol. 8, no. 4, pp. 863–894, 2019.
- [17] I. L. Markov and Y. Shi, “Simulating quantum computation by contracting tensor networks,” *SIAM Journal on Computing*, vol. 38, no. 3, pp. 963–981, 2008.
- [18] M. Rocklin, “opt_einsum: A python package for optimizing contraction order for tensor networks,” *Journal of Open Source Software*, vol. 3, no. 26, p. 753, 2018.
- [19] Y. Li, Z. Zhang, G. Song, Y. Wang, and W. Wang, “Evolutionary topology search for tensor network decomposition,” in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, pp. 6285–6294, PMLR, 2020.
- [20] Y. Li, Z. Zhang, G. Song, Y. Wang, and W. Wang, “Tensor network local search via permutation,” in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162 of *Proceedings of Machine Learning Research*, pp. 13352–13363, PMLR, 2022.
- [21] Y. Li, Z. Zhang, G. Song, Y. Wang, and W. Wang, “Tensor network architecture learning,” in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202 of *Proceedings of Machine Learning Research*, pp. 19123–19135, PMLR, 2023.
- [22] Z. Zeng, Y. Li, Z. Zhang, and W. Wang, “Bayesian tensor network structure search,” *Neural Networks*, vol. 173, pp. 106–121, 2024.

- [23] Z. Zeng, Y. Li, Z. Zhang, and W. Wang, “tngps: Discovering unknown tensor network structure with gaussian process surrogate,” in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235 of *Proceedings of Machine Learning Research*, pp. 52001–52013, PMLR, 2024.
- [24] M. Hashemizadeh, M. Liu, J. Miller, and G. Rabusseau, “Adaptive learning of tensor network structures,” *arXiv preprint*, 2021.
- [25] “Z-order curve.” https://en.wikipedia.org/wiki/Z-order_curve. Accessed 2026-02-10.
- [26] J. Miller, G. Rabusseau, and M. Hashemizadeh, “Probabilistic graphical models and tensor networks: A hybrid framework,” *arXiv preprint*, 2021.
- [27] Institute for Telecommunication Sciences, “Peak signal-to-noise ratio (psnr) algorithm description,” tech. rep., U.S. Department of Commerce, NTIA, 2009.
- [28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [30] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009.
- [31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint*, 2015.
- [32] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2 ed., 2006.
- [33] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2013.