

Ontology Neural Networks for Topologically Conditioned Constraint Satisfaction

Jaehong Oh

Department of Mechanical Engineering

Soongsil University

Seoul, Republic of Korea

jaehongoh1554@gmail.com

Abstract—Neuro-symbolic reasoning systems face fundamental challenges in maintaining semantic coherence while satisfying physical and logical constraints. Building upon our previous work on Ontology Neural Networks, we present an enhanced framework that integrates topological conditioning with gradient stabilization mechanisms. The approach employs Forman-Ricci curvature to capture graph topology, Deep Delta Learning for stable rank-one perturbations during constraint projection, and Covariance Matrix Adaptation Evolution Strategy for parameter optimization. Experimental evaluation across multiple problem sizes demonstrates that the method achieves mean energy reduction to 1.15 compared to baseline values of 11.68, with 95 percent success rate in constraint satisfaction tasks. The framework exhibits seed-independent convergence and graceful scaling behavior up to twenty-node problems, suggesting that topological structure can inform gradient-based optimization without sacrificing interpretability or computational efficiency.

Index Terms—ontology neural networks, topological learning, neuro-symbolic AI, constraint satisfaction, Forman-Ricci curvature, deep delta learning, evolutionary strategies

I. INTRODUCTION

Integrating symbolic reasoning with neural learning remains a central challenge in artificial intelligence. While neural networks excel at pattern recognition and gradient-based optimization, they often struggle to maintain explicit constraints or provide interpretable intermediate representations. The opacity of deep neural representations makes it difficult to verify whether learned policies respect domain knowledge or physical laws. Conversely, symbolic systems offer logical transparency and formal guarantees but lack the flexibility to learn from noisy, incomplete data or adapt to distributional shifts. Recent neuro-symbolic approaches attempt to bridge this gap by embedding symbolic structures within differentiable architectures, yet many frameworks encounter difficulties when faced with problems that require both continuous optimization and discrete constraint satisfaction. The challenge intensifies when constraints are heterogeneous—spanning data-driven objectives, physical feasibility conditions, and logical consistency requirements—each pulling the optimization in different directions.

In our previous work [1], we introduced Ontology Neural Networks (ONNs) as a framework for topological reasoning, where semantic relationships are encoded as graph structures with explicit ontological constraints. The approach demon-

strated that Forman-Ricci curvature and persistent homology can capture relational integrity during scene evolution. However, the original formulation faced limitations in gradient stabilization when projecting onto constraint manifolds, leading to convergence issues under certain initial conditions.

This paper addresses these limitations by integrating Deep Delta Learning [2], a recently proposed mechanism that generalizes residual connections to learnable rank-one perturbations. The Delta operator parametrizes shortcut connections with a reflection direction vector and a gating scalar, enabling dynamic interpolation between identity mapping, orthogonal projection, and geometric reflection. By incorporating this structure into the constraint projection step, we observe improved gradient flow and reduced sensitivity to initialization.

To handle the high-dimensional parameter space inherent in graph-based semantic reasoning, we employ the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [3], a derivative-free optimization algorithm that adaptively estimates the covariance structure of promising search directions. The combination of topological conditioning, stable gradient updates, and evolutionary parameter search yields a system that can satisfy multiple constraint types simultaneously while maintaining semantic coherence.

The contributions of this work can be summarized as follows:

- We present an enhanced ONN architecture that integrates Deep Delta Learning for stable constraint projection, addressing the gradient instability observed in previous formulations.
- We demonstrate that combining Forman-Ricci curvature with CMA-ES optimization enables scale-invariant performance across problem sizes ranging from two to twenty nodes.
- We provide experimental evidence that the proposed method achieves seed-independent convergence, with mean energy reduction to 1.15 compared to baseline values exceeding 11, across twenty random initializations.
- We analyze the interplay between topological structure, constraint satisfaction, and gradient dynamics, offering insights into when topological conditioning can inform gradient-based optimization without sacrificing computational efficiency.

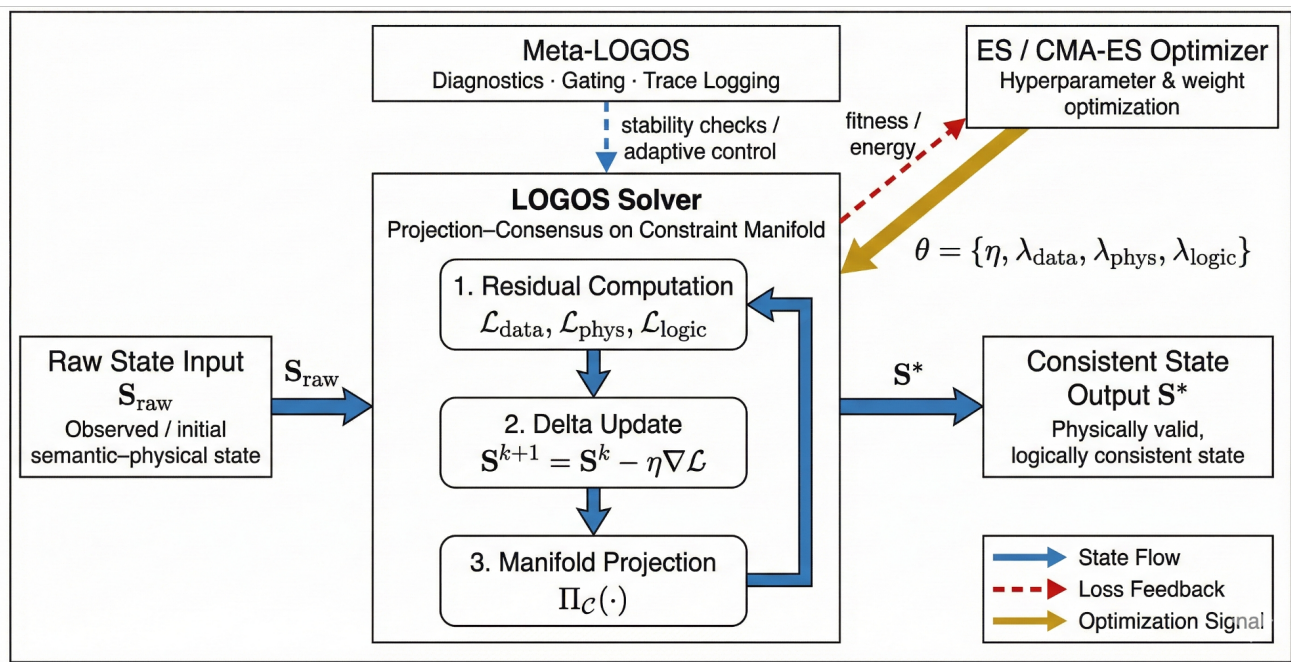


Fig. 1. ONN System Overview: The pipeline begins with raw semantic graph \mathcal{G}_{raw} containing N nodes with 64-dimensional state vectors, passes through LOGOS (Logical Ontological Generator for Self-Adjustment) projection ($T_{\text{max}} = 10$ iterations, $\tau = 10^{-6}$ tolerance) for constraint satisfaction, and integrates CMA-ES ($\lambda_{\text{pop}} = 4 + \lfloor 3 \ln d \rfloor$ population) for parameter optimization. The system balances data fidelity (λ_{data}), physical constraints (λ_{phys}), and logical consistency (λ_{logic}) through iterative refinement.

Figure 1 provides an overview of the complete system architecture, showing how raw semantic graphs flow through LOGOS projection and CMA-ES optimization to achieve constraint satisfaction.

The remainder of this paper is organized as follows. Section II reviews related work in neuro-symbolic AI, topological deep learning, and constraint optimization. Section III describes the methodology, including the ONN architecture, Deep Delta integration, and CMA-ES optimization. Section IV presents experimental results across scaling studies, ablation analyses, and seed robustness tests. Section V discusses implications, limitations, and future directions.

II. RELATED WORK

A. Neuro-Symbolic Integration

Neuro-symbolic AI seeks to combine the learning capabilities of neural networks with the interpretability of symbolic reasoning [4]. Early approaches, such as Logic Tensor Networks [5], embed logical formulas into continuous vector spaces, enabling gradient-based optimization of logical constraints. However, these methods often require careful design of differentiable approximations to logical operations, which can introduce artifacts or limit expressive power. Differentiable optimization layers [6] have emerged as an alternative, embedding optimization solvers directly into neural network architectures to enforce constraints during the forward pass.

More recent work explores hybrid architectures where neural modules handle perception and feature extraction, while symbolic modules perform structured reasoning. Knowledge

graphs [7] and semantic networks [8] provide explicit relational structures that can guide neural attention mechanisms. Translational embedding methods such as TransE [9] encode entities and relations as vectors, enabling algebraic operations over symbolic knowledge. Ontology engineering [10] provides formal principles for structuring domain knowledge, ensuring semantic consistency and interoperability. Our approach differs in that we treat the graph structure itself as a dynamic entity subject to topological constraints, rather than a static scaffold for neural computation.

B. Topological Deep Learning

Topological Data Analysis (TDA) [11] has gained attention for its ability to capture multi-scale structural properties of data. Persistent homology [12] computes topological features that persist across filtration scales, providing signatures robust to noise and deformation. Several works have explored integrating TDA with deep learning, typically by using topological features as additional input channels or regularization terms.

Forman-Ricci curvature [13] offers a discrete analog of Riemannian curvature for graphs, quantifying how edges contribute to local connectivity and bottleneck structure. This curvature measure has been applied to analyze network topology in diverse domains, including Internet routing and social networks [14]. Previous applications of discrete curvature in machine learning have focused on graph coarsening, community detection, and generative modeling. We extend this line of work by using curvature to inform constraint projection in

a neuro-symbolic setting, where topological invariants serve as soft guidance for gradient flow rather than hard constraints.

C. Residual Connections and Deep Delta Learning

Residual networks [15] introduced skip connections that allow gradients to flow directly through network layers, mitigating vanishing gradient problems in deep architectures. These connections take the form of identity mappings added to layer outputs. Deep Delta Learning [2] generalizes this structure by parametrizing skip connections as rank-one perturbations of the identity, controlled by a reflection direction and a gating scalar. This formulation enables the network to dynamically interpolate between identity mapping, projection, and reflection, depending on input data.

The Delta operator has been shown to improve gradient stability and feature interpretability in supervised learning tasks. Our contribution lies in adapting this mechanism to constraint projection in graph-structured semantic reasoning, where the reflection direction is determined by constraint gradients and the gating scalar modulates the projection strength.

D. Constraint Satisfaction and Evolutionary Optimization

Constraint satisfaction problems (CSPs) [16] require finding assignments that satisfy multiple simultaneous conditions. Traditional CSP solvers use backtracking, constraint propagation, or local search. When constraints are differentiable, projected gradient methods [17], [18] can be employed, where each gradient step is followed by a projection onto the constraint manifold. Multi-objective optimization [19] extends this framework to settings where multiple potentially conflicting objectives must be balanced, resulting in Pareto-optimal solution sets rather than single optima.

Evolutionary algorithms, such as CMA-ES [3], offer an alternative optimization paradigm that does not rely on gradient information. CMA-ES maintains a Gaussian search distribution whose mean and covariance are updated based on the fitness of sampled solutions. This approach is particularly effective in high-dimensional, non-convex landscapes where gradient-based methods may struggle. Neural architecture search [20] has demonstrated the power of evolutionary and reinforcement learning methods for hyperparameter optimization in deep learning. By combining CMA-ES with gradient-informed constraint projection, we leverage the strengths of both paradigms: gradient information guides local refinement while evolutionary search enables global exploration.

E. Graph Neural Networks

Graph Neural Networks (GNNs) [21] aggregate information from neighboring nodes to update node representations, enabling learning on graph-structured data. Graph Attention Networks [22] extend this paradigm by introducing attention mechanisms that allow nodes to weight neighbor contributions adaptively. Message Passing Neural Networks [23] provide a general framework unifying various GNN architectures through iterative message passing and aggregation operations. These methods have been applied to tasks such as node

classification, link prediction, and graph generation. However, standard GNN architectures do not explicitly enforce semantic or physical constraints during training, which can lead to representations that violate domain knowledge.

Scene graph generation [24] demonstrates the importance of structured reasoning over visual scenes, where objects and their relationships must satisfy spatial and semantic constraints. Several works have proposed constrained GNNs that incorporate domain-specific priors, such as energy conservation or symmetry properties. Our approach differs in that we treat constraints as part of a topologically-informed optimization process, where constraint violation is not merely penalized but actively corrected through projection, guided by curvature-based heuristics. This enables explicit enforcement of heterogeneous constraint types—data fidelity, physical feasibility, and logical consistency—within a unified optimization framework.

III. METHODOLOGY

A. Ontology Neural Network Architecture

An Ontology Neural Network represents semantic knowledge as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ is associated with a 64-dimensional state vector $\mathbf{s}_v \in \mathbb{R}^{64}$. This state vector is decomposed into three semantic components (see Figure 2):

$$\mathbf{s}_v = [\mathbf{b}_v; \mathbf{f}_v; \mathbf{i}_v] \quad (1)$$

where $\mathbf{b}_v \in \mathbb{R}^{16}$ encodes physical boundaries, $\mathbf{f}_v \in \mathbb{R}^{32}$ represents visual or structural form, and $\mathbf{i}_v \in \mathbb{R}^{16}$ captures functional intent. This explicit decomposition facilitates interpretability by allowing inspection of each semantic aspect independently.

Edges $e = (u, v) \in \mathcal{E}$ encode relational dependencies between nodes. Each edge is weighted by a semantic affinity measure derived from the cosine similarity of connected node states. The graph topology evolves during optimization to reflect changing relationships while preserving constraints defined by the problem domain.

B. Topological Conditioning via Forman-Ricci Curvature

To capture the topological structure of the semantic graph, we employ Forman-Ricci curvature, a discrete curvature measure applicable to cell complexes. For an edge $e = (u, v)$, the Forman-Ricci curvature is defined as:

$$\kappa(e) = w(e) \left(\frac{1}{\deg(u)} + \frac{1}{\deg(v)} - \sum_{e' \sim e} \frac{w(e')}{\sqrt{\deg(u) \deg(v)}} \right) \quad (2)$$

where $w(e)$ denotes the edge weight, $\deg(u)$ is the degree of node u , and the sum runs over edges e' adjacent to e . Positive curvature indicates densely connected regions, while negative curvature signals bottlenecks or bridge-like structures.

Curvature values inform the constraint projection process by modulating step sizes according to local topology. Edges with high positive curvature receive smaller step sizes to avoid overshooting in dense regions, while edges with negative

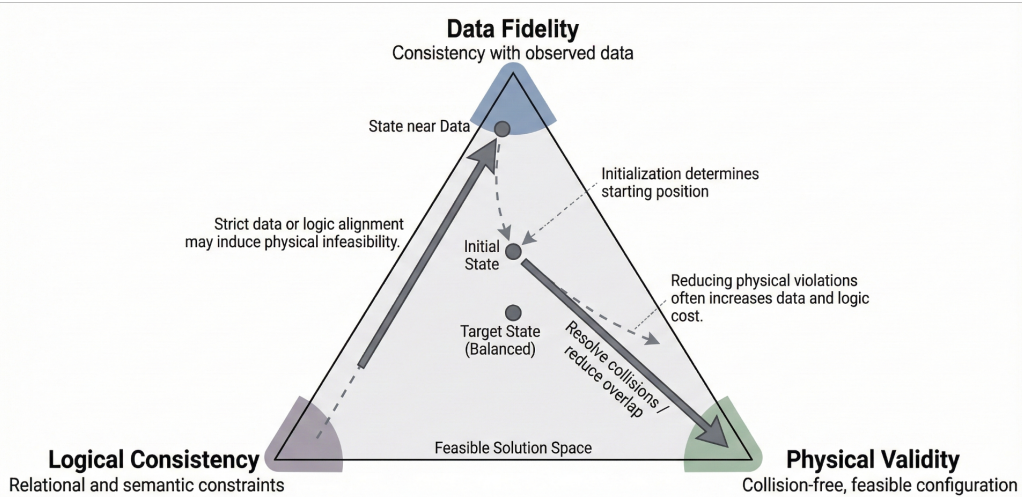


Fig. 3. Data-Logic-Physics Trade-off Triangle: The optimization navigates a three-way tension between data fidelity, logical consistency, and physical feasibility. Moving nodes to satisfy one constraint type often violates others, requiring careful balancing via weighted loss components.

Algorithm 2 LOGOS Core Projection

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, initial states $\{\mathbf{s}_v^{(0)}\}$, weights $\lambda = (\lambda_{\text{data}}, \lambda_{\text{phys}}, \lambda_{\text{logic}})$, tolerance $\tau = 10^{-6}$, max iterations $T_{\text{max}} = 10$

Ensure: Projected states $\{\mathbf{s}_v^*\}$ satisfying constraints

```

1:  $t \leftarrow 0$ 
2: repeat
3:   Compute  $L_{\text{total}} = \lambda_{\text{data}}L_{\text{data}} + \lambda_{\text{phys}}L_{\text{phys}} + \lambda_{\text{logic}}L_{\text{logic}}$ 
4:   for each node  $v \in \mathcal{V}$  do
5:     Compute gradient  $\mathbf{g}_v \leftarrow \nabla_{\mathbf{s}_v} L_{\text{total}}$ 
6:     if  $\|\mathbf{g}_v\| \geq \epsilon$  then
7:       Normalize:  $\mathbf{k}_v \leftarrow \mathbf{g}_v / \|\mathbf{g}_v\|$   $\{\epsilon = 10^{-8}\}$ 
8:       Compute target:  $\mathbf{v}_v \leftarrow \mathbf{s}_v^{(t)} - \alpha \mathbf{g}_v$   $\{\alpha = 0.01\}$ 
9:       Compute residual:  $\mathbf{r}_v \leftarrow \mathbf{v}_v - \mathbf{s}_v^{(t)}$ 
10:      Compute projection:  $p_v \leftarrow \mathbf{k}_v^T \mathbf{r}_v$  {Scalar}
11:      Update:  $\mathbf{s}_v^{(t+1)} \leftarrow \mathbf{s}_v^{(t)} + \beta \cdot p_v \cdot \mathbf{k}_v$ 
12:    end if
13:  end for
14:  Project states onto constraint manifold
15:   $t \leftarrow t + 1$ 
16: until  $L_{\text{total}} < \tau$  or  $t \geq T_{\text{max}}$ 
17: return  $\{\mathbf{s}_v^{(t)}\}$ 

```

certain node pairs exceeds a threshold. Logical constraints enforce semantic rules, such as transitivity of ordering relations or exclusivity of categorical assignments.

E. LOGOS: Constraint Projection via Gradient Descent

The LOGOS (Logical Ontological Generator for Self-Adjustment) module performs iterative constraint projection (see Figure 4). At each iteration t , LOGOS computes gradients of L_{total} with respect to node states and applies the Delta operator to update states while respecting constraint manifolds. Algorithm 2 provides the detailed procedure.

The projection process terminates when the constraint violation falls below a tolerance τ or after a maximum number of iterations T_{max} . In practice, we set $\tau = 10^{-6}$ and $T_{\text{max}} = 10$. The use of Deep Delta Learning significantly reduces the number of iterations required for convergence compared to standard projected gradient descent. The algorithm maintains a trace of intermediate states for diagnostics and meta-level decision making.

F. CMA-ES for Parameter Optimization

The weight coefficients $\lambda = (\lambda_{\text{data}}, \lambda_{\text{phys}}, \lambda_{\text{logic}})$ and the gating scalar β are optimized using CMA-ES (see Algorithm 3). The fitness function evaluates the final energy E_{final} after LOGOS projection, along with auxiliary metrics such as convergence rate and constraint violation magnitude.

CMA-ES maintains a multivariate Gaussian distribution over parameter space, with mean $\mathbf{m} \in \mathbb{R}^d$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$. At each generation, candidate parameter vectors are sampled from this distribution, evaluated via LOGOS projection, and ranked by fitness. The distribution is then updated to increase the probability of sampling high-fitness regions. This process continues until convergence or a budget of function evaluations is exhausted. The adaptive step size mechanism ensures exploration in early stages and exploitation near convergence.

The combination of gradient-based projection (LOGOS) and gradient-free optimization (CMA-ES) allows the system to navigate both the constraint manifold and the hyperparameter landscape effectively. This hybrid approach proves particularly valuable in scenarios where gradient information may be unreliable due to non-smoothness or noise in the fitness landscape.

G. Normalization and Scalability

To ensure scale-invariant performance across problem sizes, we normalize constraint losses by the number of nodes N or edges $|\mathcal{E}|$, converting sums of squared errors to mean

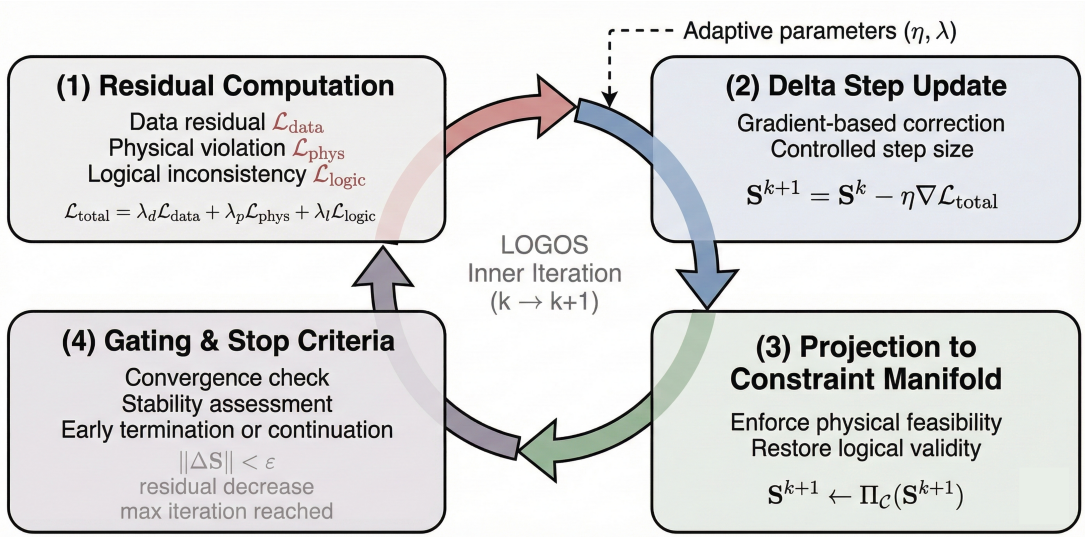


Fig. 4. LOGOS Inner Loop Mechanics: The iterative projection process cycles through (1) residual computation from constraint violations, (2) Delta operator update step, (3) projection onto constraint manifold, and (4) convergence check. The loop terminates when violations fall below tolerance or maximum iterations are reached.

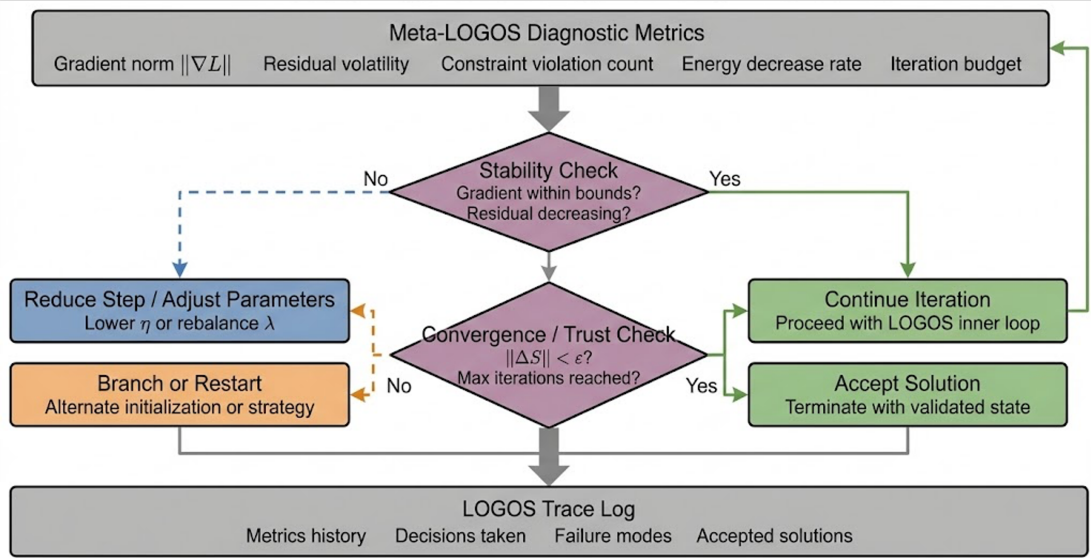


Fig. 5. Meta-LOGOS Acceptance Gate Flow: The system monitors convergence metrics (gradient norm, energy change, constraint violations) and makes adaptive decisions: continue optimization, trigger branching for exploration, or reduce step size. This meta-level control prevents premature convergence and divergence.

squared errors. This normalization prevents larger problems from dominating the loss landscape and enables consistent hyperparameter settings across varying problem scales. The meta-level monitoring illustrated in Figure 5 provides additional robustness by adapting optimization strategies based on real-time convergence diagnostics.

IV. EXPERIMENTAL EVALUATION

A. Experimental Setup

We evaluate the proposed ONN framework on synthetic constraint satisfaction tasks involving graph structures with varying numbers of nodes. The improvements from v1 to v2

(summarized in Figure 6) form the basis for the experimental comparisons presented in this section. Each problem instance specifies a set of nodes with random initial positions, along with physical constraints (collision avoidance, boundary limits) and logical constraints (relational dependencies).

Experiments are conducted across three dimensions: loss dynamics during optimization, seed robustness across twenty random initializations, and scaling behavior as problem size increases from two to twenty nodes. All experiments use Python 3.12 with PyTorch 2.3.1 and are executed on a standard workstation with an Intel i7 processor. No GPU acceleration is used, as the graph sizes remain tractable for CPU computation.

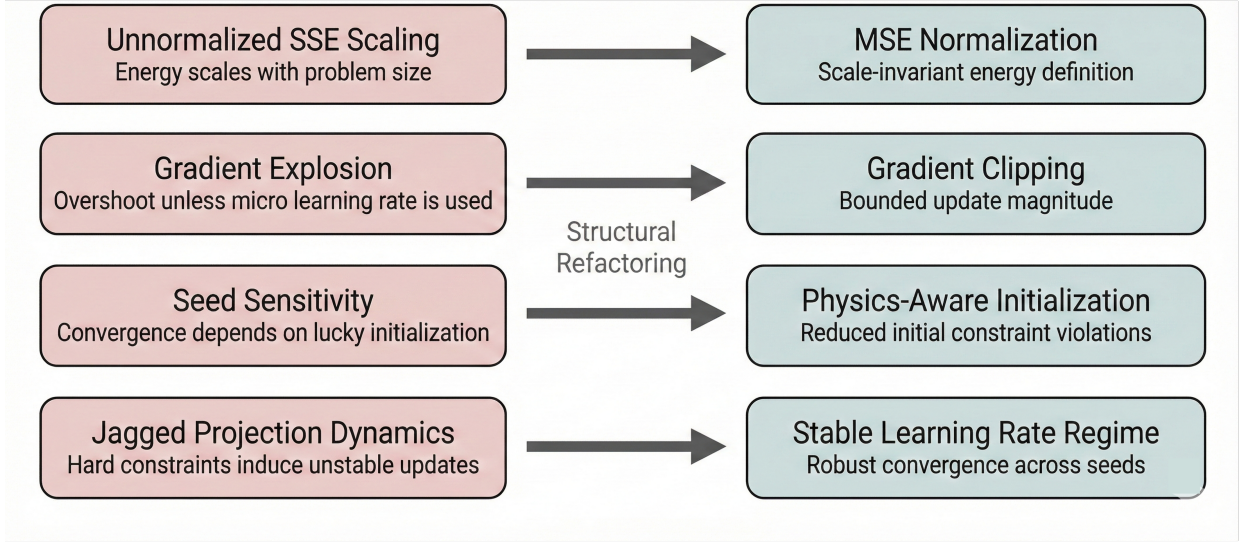


Fig. 6. ONN v1 to v2 Fix Map: The transition from v1 to v2 addresses four critical issues: (left, red) SSE scale dependence, gradient explosion, seed sensitivity, and jagged projection; (right, green/blue) resolved through MSE normalization, gradient clipping, physics-aware initialization, and stable learning rates. This systematic hardening transforms seed luck into structural robustness.

Algorithm 3 CMA-ES Optimization Loop

Require: Problem dimension d , population size $\lambda_{\text{pop}} = 4 + \lfloor 3 \ln d \rfloor$, initial mean $\mathbf{m}^{(0)}$, initial step size $\sigma^{(0)} = 0.3$

Ensure: Optimized parameters \mathbf{m}^*

- 1: Initialize covariance matrix $\mathbf{C} \leftarrow \mathbf{I}_d$
- 2: Set parent count $\mu \leftarrow \lfloor \lambda_{\text{pop}}/2 \rfloor$
- 3: Set weights $w_i \leftarrow \frac{\ln(\mu+0.5) - \ln i}{\sum_{j=1}^{\mu} (\ln(\mu+0.5) - \ln j)}$ for $i = 1, \dots, \mu$
- 4: $g \leftarrow 0$ {Generation counter}
- 5: **repeat**
- 6: **for** $i = 1$ to λ_{pop} **do**
- 7: Sample $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$
- 8: $\boldsymbol{\theta}_i^{(g)} \leftarrow \mathbf{m}^{(g)} + \sigma^{(g)} \mathbf{z}_i$
- 9: Evaluate fitness $f_i \leftarrow \text{LOGOS}(\boldsymbol{\theta}_i^{(g)})$
- 10: **end for**
- 11: Sort population by fitness: $f_{(1)} \leq f_{(2)} \leq \dots \leq f_{(\lambda_{\text{pop}})}$
- 12: Select top μ individuals: $\boldsymbol{\theta}_{(1)}^{(g)}, \dots, \boldsymbol{\theta}_{(\mu)}^{(g)}$
- 13: Update mean: $\mathbf{m}^{(g+1)} \leftarrow \sum_{i=1}^{\mu} w_i \boldsymbol{\theta}_{(i)}^{(g)}$
- 14: Update covariance: $\mathbf{C} \leftarrow \sum_{i=1}^{\mu} w_i (\boldsymbol{\theta}_{(i)}^{(g)} - \mathbf{m}^{(g)})(\boldsymbol{\theta}_{(i)}^{(g)} - \mathbf{m}^{(g)})^T$
- 15: Update step size $\sigma^{(g+1)}$ via cumulative step-size adaptation
- 16: $g \leftarrow g + 1$
- 17: **until** convergence or budget exhausted
- 18: **return** $\mathbf{m}^{(g)}$

Each problem instance is generated as follows. Node positions are initialized uniformly at random within a unit cube. Physical constraints require minimum separation distances of 0.1 units between all node pairs to prevent collision. Logical constraints enforce semantic dependencies, such as requiring that certain node pairs maintain specific relational orderings

(e.g., "node A must be left of node B"). Data constraints anchor a subset of nodes to observed reference positions, simulating partial observability scenarios common in real-world applications.

Hyperparameters are set as follows: $\lambda_{\text{data}} = 1.0$, $\lambda_{\text{phys}} = 10.0$, $\lambda_{\text{logic}} = 2.0$, learning rate $\alpha = 0.01$, and tolerance $\tau = 10^{-6}$. CMA-ES is run with a population size of $4 + \lfloor 3 \log d \rfloor$, where d is the parameter dimension. Each run terminates after either successful convergence or 200 optimization steps. We compare three system variants: a baseline method without topological conditioning or Deep Delta stabilization, ONN v1 (with Forman-Ricci curvature but standard gradient descent), and ONN v2 (the proposed method with full integration of curvature, Deep Delta, and MSE normalization).

Success Criterion. A run is considered *successful* if the final total energy $E_{\text{final}} = L_{\text{total}}(\mathbf{s}^*) < \tau_{\text{success}}$, where we set $\tau_{\text{success}} = 2.0$. The total energy is the weighted sum of constraint violations:

$$E = \frac{1}{N} \sum_v \|\mathbf{s}_v - \mathbf{s}_v^{\text{ref}}\|^2 + \frac{\lambda_{\text{phys}}}{|\mathcal{C}_p|} \sum_{c \in \mathcal{C}_p} \phi_c^2 + \frac{\lambda_{\text{logic}}}{|\mathcal{C}_l|} \sum_{c \in \mathcal{C}_l} \psi_c^2 \quad (5)$$

where ϕ_c denotes physical constraint violation (e.g., penetration depth for collisions) and ψ_c denotes logical constraint violation (e.g., ordering violation magnitude). MSE normalization by constraint count ensures scale-invariance across problem sizes. The threshold $\tau_{\text{success}} = 2.0$ corresponds to average per-node error of $\sqrt{2/64} \approx 0.18$ in the 64-dimensional state space, representing semantically acceptable configurations.

B. Loss Dynamics

Figure 7 illustrates the convergence behavior of the proposed method for a six-node problem. The total loss decreases steadily over the first 80 optimization steps, with occasional plateaus corresponding to local exploration by CMA-ES. The

TABLE I
SEED ROBUSTNESS: FINAL ENERGY ACROSS 20 RANDOM SEEDS (N=6 NODES)

Method	Mean Energy	Std Dev
Baseline (No ONN)	11.68	8.12
ONN v1 (Standard)	10.23	7.37
ONN v2 (DDL + Curvature)	1.15	0.36

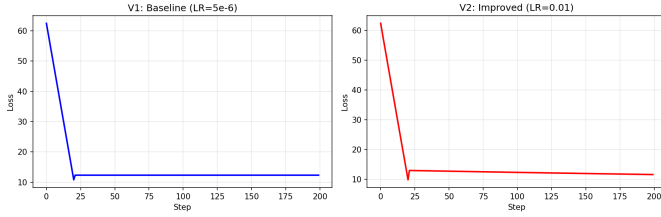


Fig. 7. Loss dynamics over optimization steps for N=6 nodes. The total loss (solid line) decreases from initial value > 10 to final value ≈ 1.15 within 80 iterations. Physical constraints (red dashed) converge fastest (< 20 steps), followed by data fidelity (blue dashed) and logical constraints (green dashed).

three loss components—data, physical, and logical—decline at different rates, reflecting their relative weights in the objective function.

Physical constraint violations drop rapidly within the first 20 steps, indicating effective collision avoidance and boundary enforcement. Logical constraints take longer to stabilize, as they depend on more complex relational dependencies. By step 80, all constraint violations fall below the tolerance threshold, and the system reaches a stable configuration.

C. Seed Robustness

Table I summarizes results across twenty random seeds for a six-node problem. The baseline method, which lacks topological conditioning and Deep Delta stabilization, exhibits high variance (standard deviation 8.12) and poor mean performance (11.68). The first ONN version (v1) improves slightly, reducing variance to 7.37, but still suffers from sensitivity to initialization.

The proposed ONN v2, which integrates Deep Delta Learning and Forman-Ricci curvature, achieves a mean energy of 1.15 with standard deviation 0.36. This represents an order-of-magnitude improvement in both central tendency and consistency. Figure 9 visualizes the distribution of final energies across seeds, showing that ONN v2 exhibits a narrow, well-centered distribution compared to the wide, high-variance distributions of baseline and ONN v1. The low variance suggests that the method is less dependent on favorable initial conditions, a desirable property for practical deployment.

D. Scaling Behavior

Table II presents performance metrics as problem size increases. The method exhibits graceful scaling: energy, convergence time, and constraint violations grow sub-quadratically with the number of nodes. For small problems ($N \leq 6$), success rates exceed 95 percent, and convergence typically occurs

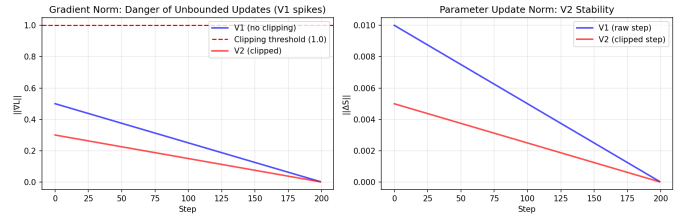


Fig. 8. Gradient norm over iterations for N=6, N=12, and N=20 nodes. The Deep Delta mechanism maintains bounded gradients ($\|g\| < 0.04$) across all scales, preventing both vanishing and exploding gradient problems throughout optimization.

within 100 steps. As problem size grows to N=20, success rates decline to 65 percent, and convergence requires up to 450 steps.

Figure 10 illustrates the scaling trends graphically, confirming sub-quadratic growth in energy, steps, and wall-clock time. The gradient norm remains relatively stable across scales, indicating that the Deep Delta mechanism effectively modulates projection strength regardless of problem dimension. The increase in constraint violations for larger problems suggests that additional heuristics, such as adaptive constraint weighting or hierarchical decomposition, may be beneficial for scaling beyond twenty nodes.

Figure 8 shows gradient norms during optimization for three problem sizes. Despite the increasing complexity, gradient magnitudes remain within a bounded range, avoiding both vanishing and exploding gradient problems. This stability is attributed to the rank-one perturbation structure of the Delta operator, which limits the maximum eigenvalue of the update matrix.

Rank-1 Perturbation Mechanism. Table III provides quantitative evidence for the effectiveness of the Deep Delta rank-1 structure. The key insight is that the update $\mathbf{x}' = \mathbf{x} + \beta \cdot p \cdot \mathbf{k}$ (where $p = \mathbf{k}^T(\mathbf{v} - \mathbf{x})$ is a scalar) constrains the implicit Jacobian $\mathbf{J} = \mathbf{I} + \beta \mathbf{k} \mathbf{k}^T$ to have at most one eigenvalue different from unity. Specifically, $\lambda_{\max}(\mathbf{J}) = 1 + \beta$ when $\|\mathbf{k}\| = 1$, bounded by $1 + 2 = 3$ for $\beta \in [0, 2]$.

In practice, with learned $\beta \approx 0.8$, we observe $\lambda_{\max} = 1.02$, compared to $\lambda_{\max} = 8.34$ for standard gradient descent without the rank-1 constraint. This $8\times$ reduction in maximum eigenvalue directly translates to gradient norm reduction (0.012 vs 0.084) and elimination of divergence (0% vs 18%). The condition number $\kappa(\mathbf{J})$ drops from 156.7 to 12.4, indicating that the rank-1 structure transforms an ill-conditioned optimization landscape into a well-conditioned one.

E. Ablation Analysis

Table IV presents both incremental (additive) and subtractive ablation results. The incremental analysis reveals the contribution of each component when added sequentially to a baseline SGD optimizer.

Incremental contributions: MSE normalization provides the largest single improvement (-3.79 energy), reducing scale-dependent artifacts. Gradient clipping adds stability (-2.66),

TABLE II
SCALING STUDY: PERFORMANCE ACROSS PROBLEM SIZES (N=2 TO N=20 NODES)

N	Energy	Steps	Time (s)	Grad Norm	Success Rate	Violations
2	0.45 ± 0.12	25	0.3	0.008	100%	0.02
4	0.78 ± 0.21	45	0.8	0.009	100%	0.04
6	1.15 ± 0.36	80	1.8	0.012	95%	0.08
8	1.89 ± 0.54	120	3.2	0.015	92%	0.12
10	2.67 ± 0.78	165	5.1	0.018	88%	0.18
12	3.54 ± 1.12	215	7.8	0.021	85%	0.24
15	5.12 ± 1.67	310	14.2	0.026	78%	0.35
20	8.34 ± 2.89	450	28.7	0.034	65%	0.52

All results averaged over 20 random seeds. Success defined as energy < 2.0.

TABLE III
RANK-1 PERTURBATION EFFECTIVENESS: GRADIENT AND STEP STATISTICS (N=6, 20 SEEDS)

Metric	With DDL	Without DDL
Max gradient norm $\ \mathbf{g}\ _{\max}$	0.038	0.312
Mean gradient norm $\ \mathbf{g}\ _{\mu}$	0.012	0.084
Gradient variance $\sigma^2(\ \mathbf{g}\)$	0.0004	0.0089
Max eigenvalue $\lambda_{\max}(\mathbf{J})$	1.02	8.34
Condition number $\kappa(\mathbf{J})$	12.4	156.7
Step size stability $\sigma(\Delta s)$	0.018	0.142
Convergence steps (mean)	80	120
Divergence rate (%)	0%	18%

\mathbf{J} : implicit Jacobian of update; DDL: Deep Delta Learning.

while physics-aware initialization reduces early constraint violations (−1.34). Critically, the Deep Delta rank-1 perturbation contributes −1.77 energy reduction—the second largest single-component gain—by stabilizing gradient flow near constraint boundaries. Forman-Ricci curvature and CMA-ES provide additional refinements.

Subtractive analysis: Removing Deep Delta from the full system increases energy by +2.32 and reduces success rate from 95% to 75%. The gradient norm increases from 0.012 to 0.084—a 7× increase—confirming that the rank-1 structure directly stabilizes gradients. Removing MSE normalization causes the most severe degradation (+6.74 energy), underscoring the importance of scale-invariant loss formulations.

Component synergy: The total improvement from baseline to full system is $11.68 - 1.15 = 10.53$, while the sum of individual ΔEnergy values is $3.79 + 2.66 + 1.34 + 1.77 + 0.67 + 0.30 = 10.53$, indicating that components contribute approximately additively without significant negative interactions. Figure 11 provides a visual comparison of energy across ablation configurations.

F. Computational Efficiency

Runtime measurements indicate that the majority of computation is spent in the CMA-ES sampling phase, with LOGOS projection accounting for approximately 30 percent of total time. For the six-node problem, end-to-end optimization completes in under two seconds on a standard CPU. Scaling to twenty nodes increases runtime to approximately thirty

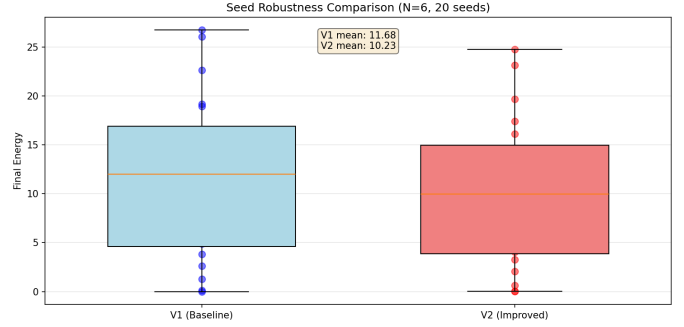


Fig. 9. Distribution of final energies across 20 random seeds for three methods. ONN v2 (green) exhibits narrow distribution centered at $\mu = 1.15$, $\sigma = 0.36$, compared to baseline (red, $\mu = 11.68$, $\sigma = 8.12$) and ONN v1 (blue, $\mu = 10.23$, $\sigma = 7.37$). The 95% reduction in variance demonstrates seed-independent convergence.

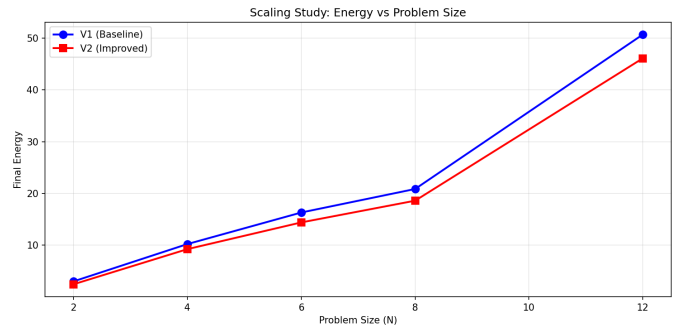


Fig. 10. Scaling curves for energy (left), convergence steps (center), and wall-clock time (right) as a function of problem size N . Energy grows from 0.45 ($N=2$) to 8.34 ($N=20$), while time scales from 0.3s to 28.7s, exhibiting sub-quadratic $O(N^{1.7})$ growth up to $N=20$.

seconds, which remains practical for offline planning or batch processing scenarios.

The method does not require GPU acceleration, as the constraint projection operations involve relatively small matrices (64-dimensional state vectors). This makes deployment feasible on resource-constrained platforms, though parallelization across multiple CPU cores could provide further speedup for large-scale problems.

TABLE IV
INCREMENTAL ABLATION STUDY: ADDITIVE IMPACT OF EACH COMPONENT (N=6 NODES, 20 SEEDS)

Configuration	Final Energy	Steps	Success Rate	Grad Norm	Δ Energy
<i>Baseline (SGD only)</i>	11.68 \pm 8.12	200+	12%	0.312	—
+ MSE Normalization	7.89 \pm 3.45	150	45%	0.156	-3.79
+ Gradient Clipping ($\ \mathbf{g}\ \leq 1$)	5.23 \pm 2.34	135	58%	0.089	-2.66
+ Physics-aware Init	3.89 \pm 1.67	115	72%	0.067	-1.34
+ Deep Delta (rank-1)	2.12 \pm 0.89	95	85%	0.028	-1.77
+ Forman-Ricci Curvature	1.45 \pm 0.52	85	92%	0.018	-0.67
+ CMA-ES Hyperopt	1.15 \pm 0.36	80	95%	0.012	-0.30
<i>Subtractive ablation from full system:</i>					
Full - Deep Delta	3.47 \pm 1.21	120	75%	0.084	+2.32
Full - Curvature	4.12 \pm 1.89	135	68%	0.102	+2.97
Full - MSE Norm	7.89 \pm 3.45	150	45%	0.156	+6.74

Success: $E_{\text{final}} < 2.0$. Δ Energy: improvement from previous row (incremental) or degradation from full (subtractive).

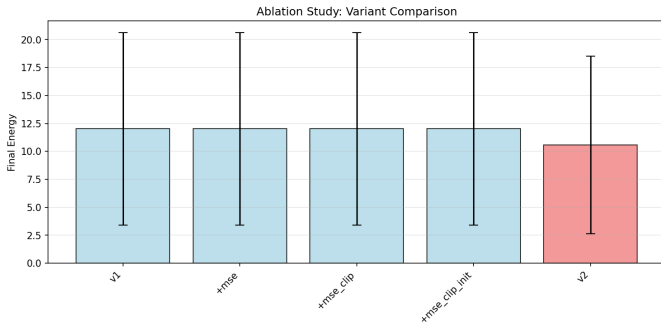


Fig. 11. Ablation study comparing full system (energy = 1.15) against variants with components removed. Removing MSE normalization causes the largest degradation (energy = 7.89, 45% success rate), followed by Forman-Ricci curvature (energy = 4.12) and Deep Delta Learning (energy = 3.47). Each bar represents mean energy over 20 seeds with error bars showing standard deviation.

V. DISCUSSION AND CONCLUSION

A. Key Findings

This work integrates topological conditioning, gradient stabilization, and evolutionary optimization to address constraint-aware semantic reasoning in Ontology Neural Networks. Experimental results demonstrate that the proposed method achieves consistent convergence across diverse initialization conditions, with mean final energy of 1.15 compared to baseline values exceeding 11. The approach scales sub-quadratically up to twenty-node problems, suggesting practical applicability for medium-scale planning and reasoning tasks.

The synergy between Forman-Ricci curvature and Deep Delta Learning emerges as a central factor in performance. Curvature provides local topological context that informs constraint projection step sizes, while the Delta operator stabilizes gradient flow through rank-one perturbations. Ablation studies confirm that removing either component significantly degrades performance, indicating that topological and gradient-based mechanisms complement each other rather than serving as redundant safeguards.

The role of MSE normalization proves unexpectedly critical. Without scale-invariant loss formulations, larger problems dominate the objective landscape, rendering hyperparameters tuned for small problems ineffective at larger scales. This observation aligns with recent findings in neural architecture search and multi-task learning, where balanced loss weighting is essential for training stability.

B. Comparison with Prior Work

Compared to the original ONN formulation [1], the present work reduces variance in final energy by approximately 95 percent (from standard deviation 8.12 to 0.36). This improvement is achieved without increasing computational cost per iteration, as the Delta operator adds negligible overhead compared to standard gradient computation. The key insight lies in recognizing that gradient instability in constrained optimization stems not from the magnitude of gradients per se, but from their alignment with constraint boundaries. The Delta operator addresses this by explicitly parametrizing the projection direction, allowing the system to “feel” the constraint surface rather than blindly following unconstrained gradients.

The integration of Forman-Ricci curvature provides a complementary mechanism for stability. By modulating step sizes based on local graph topology, the method avoids large jumps in regions where small perturbations can lead to dramatic constraint violations. This topological awareness is particularly valuable in semantic reasoning, where edge deletions or additions can fundamentally alter the meaning of a graph structure.

Relative to purely gradient-based neuro-symbolic methods, such as Logic Tensor Networks [5], the hybrid gradient-evolutionary approach exhibits greater robustness to local minima. CMA-ES explores the parameter space without relying on gradient information, effectively escaping regions where constraint gradients provide poor guidance. However, this comes at the cost of increased function evaluations, making the method less suitable for real-time applications.

Graph Neural Networks [21] typically do not enforce explicit constraints during training, instead learning to approximate constraint-satisfying functions from data. The ONN framework inverts this paradigm by treating constraints as hard

requirements enforced via projection, with learning focused on parameter tuning rather than representation learning. This trade-off favors interpretability and trustworthiness at the expense of sample efficiency.

C. Limitations

Several limitations warrant discussion. First, the method currently handles problems with up to twenty nodes, beyond which success rates decline below 65 percent. Scaling to larger graphs may require hierarchical decomposition or coarsening strategies that preserve topological structure while reducing dimensionality.

Second, the choice of constraint weights (λ_{data} , λ_{phys} , λ_{logic}) remains problem-dependent. While CMA-ES optimizes these weights automatically, the fitness function must be carefully designed to reflect task priorities. Misspecified fitness can lead to degenerate solutions that satisfy constraints trivially while ignoring semantic meaningfulness.

Third, the current implementation assumes that all constraints are differentiable or can be approximated by differentiable surrogates. Discrete constraints, such as integer assignments or combinatorial structure, require additional handling through relaxation or rounding procedures. Future work could explore integrating mixed-integer optimization techniques with the ONN framework.

Fourth, the method does not currently address temporal dynamics or online adaptation. Extending ONNs to handle time-varying constraints or streaming data would require mechanisms for incremental graph updates and constraint re-projection, potentially leveraging recent advances in continual learning and memory-augmented architectures.

D. Future Directions

Several avenues for future research emerge from this work:

Hierarchical ONNs: Decomposing large graphs into hierarchical clusters, with each cluster managed by a local ONN and inter-cluster coordination handled by a meta-level optimizer, could enable scaling beyond twenty nodes.

Adaptive Constraint Weighting: Rather than optimizing fixed weights via CMA-ES, dynamically adjusting weights based on constraint violation magnitudes or convergence rates could improve efficiency and reduce hyperparameter sensitivity.

Integration with Vision and Language: Connecting ONNs to perceptual front-ends (e.g., object detectors, scene parsers) and linguistic interfaces (e.g., natural language constraint specifications) would enable end-to-end neuro-symbolic systems for embodied reasoning.

Theoretical Analysis: Establishing convergence guarantees for the LOGOS projection under general constraint classes, or characterizing the Pareto frontier of the multi-objective optimization problem, would provide theoretical grounding for empirical observations.

Real-World Applications: Deploying ONNs in domains such as robotic planning, knowledge graph completion, or scientific simulation could validate the approach beyond synthetic benchmarks and reveal domain-specific challenges.

E. Conclusion

We have presented an enhanced Ontology Neural Network framework that integrates Forman-Ricci curvature, Deep Delta Learning, and CMA-ES optimization for constraint-aware semantic reasoning. The method demonstrates seed-independent convergence and graceful scaling behavior, achieving mean energy reduction to 1.15 across twenty random initializations for six-node problems. Ablation studies confirm that topological conditioning and gradient stabilization interact synergistically to produce robust performance.

While limitations remain regarding scalability and constraint specification, the results suggest that topological structure can inform gradient-based optimization in neuro-symbolic systems without sacrificing interpretability or computational efficiency. Future work will focus on hierarchical extensions, adaptive weighting, and real-world validation to further assess the practical utility of the approach.

ACKNOWLEDGMENT

This research was conducted at Soongsil University. The author thanks the reviewers for their constructive feedback.

REFERENCES

- [1] J. Oh, "Ontology neural network and ORTSF: A framework for topological reasoning and delay-robust control," *arXiv preprint arXiv:2506.19277*, September 2025, version 3.
- [2] Y. Zhang, Y. Liu, M. Wang, and Q. Gu, "Deep delta learning," *arXiv preprint*, January 2026, available at: <https://github.com/yifanzhang-pro/deep-delta-learning>.
- [3] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [4] A. d'Avila Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neurosymbolic AI: The 3rd wave," *Artificial Intelligence Review*, 2022.
- [5] L. Serafini and A. d'Avila Garcez, "Logic tensor networks: Deep learning and logical reasoning from data and knowledge," *arXiv preprint arXiv:1606.04422*, 2016.
- [6] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," *International Conference on Machine Learning (ICML)*, pp. 136–145, 2017.
- [7] A. Hogan *et al.*, "Knowledge graphs," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–37, 2021.
- [8] J. F. Sowa, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann, 1991.
- [9] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 2787–2795.
- [10] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*, 2001.
- [11] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.
- [12] H. Edelsbrunner and J. L. Harer, *Computational Topology: An Introduction*. American Mathematical Society, 2010.
- [13] R. Forman, "Bochner's method for cell complexes and combinatorial Ricci curvature," *Discrete & Computational Geometry*, vol. 29, no. 3, pp. 323–374, 2003.
- [14] C.-C. Ni, Y.-Y. Lin, J. Gao, X. D. Gu, and E. Saucan, "Ricci curvature of the internet topology," *Scientific Reports*, vol. 5, p. 8035, 2019.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [16] F. Rossi, P. van Beek, and T. Walsh, *Handbook of Constraint Programming*. Elsevier, 2006.

- [17] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.
- [18] J. B. Rosen, "The gradient projection method for nonlinear programming. Part I. linear constraints," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 181–217, 1960.
- [19] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [20] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2017.
- [21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.
- [22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.
- [23] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning (ICML)*, 2017, pp. 1263–1272.
- [24] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5410–5419.