

# Research Proposal: Exploring Chiral Topologies for Enhanced Gradient Descent

Paul Lowndes  
ZeroTrust@NSHkr.com

December 4, 2024

## Abstract

This research proposal outlines a novel approach to gradient descent optimization, termed Chiral Gradient Descent (CGD), which incorporates topological information and rotational dynamics inspired by chirality in biological systems. The proposed methodology will investigate how chiral structures within neural networks can be leveraged to improve exploration of the parameter space, potentially leading to more robust and efficient training. The research will involve developing a mathematical framework for CGD, implementing the algorithm in a deep learning framework, and conducting experiments on benchmark datasets to evaluate CGD's performance compared to standard gradient descent methods. The expected outcomes include a mathematical formulation of CGD, an open-source implementation, and experimental results demonstrating the potential benefits of CGD for various machine learning tasks.

## 1 Introduction

Gradient descent, while a cornerstone of machine learning, often struggles to escape local minima, particularly in complex, high-dimensional loss landscapes. This limitation hinders the training of deep learning models, especially in challenging real-world applications. To address this, we propose **Chiral Gradient Descent (CGD)**, a novel optimization approach incorporating topological information and rotational dynamics inspired by chirality in biological systems. We hypothesize that CGD will enhance exploration of the parameter space, leading to more robust and efficient training, faster convergence, and improved performance for models with inherent chiral properties. This proposal outlines the mathematical framework for CGD, details the planned methodology, and discusses anticipated outcomes.

## 2 Chiral Gradient Descent

Chiral Gradient Descent (CGD) modifies the gradient update rule by incorporating chiral vectors, which introduce rotational dynamics into the optimization process. This approach is inspired by natural asymmetry observed in biological systems. The chiral

term, incorporating a sigmoid function, allows for dynamic adjustments based on topological distances within the network, potentially enhancing exploration of the parameter space and leading to more robust convergence. This sigmoid function modulates the influence of each chiral pair based on the topological distance between neurons, allowing for local chiral effects to dominate while diminishing the impact of distant pairs. The mathematical formulation of CGD involves the cross product of the gradient with a chiral vector, adding a layer of complexity and potential to the optimization process.

### 3 Applying Chiral Topologies

We represent a neural network’s topology as a graph  $G = (V, E)$ . A chiral pair of neurons  $(v_i, v_j)$  is defined based on topological asymmetry. This asymmetry is quantified using the difference in shortest path lengths from a common ancestor node to  $v_i$  and  $v_j$ , capturing the relative "distance" of each node from their shared history or context within the network. Larger path differences indicate a greater degree of asymmetry, suggesting a stronger chiral relationship. Other topological features, such as local curvature of the loss landscape, or the difference in the densities of the neighborhoods surrounding  $v_i$  and  $v_j$  relative to some common ancestor, could also be incorporated into this asymmetry calculation.

For each chiral pair  $(v_i, v_j)$ , a chiral vector  $\mathbf{c}_{ij}$  is defined in the parameter space. The direction of  $\mathbf{c}_{ij}$  corresponds to the direction in parameter space that maximizes the difference in the gradients of the loss function with respect to the parameters associated with nodes  $v_i$  and  $v_j$ . This direction represents the axis around which the chiral rotation will occur during the gradient descent update. The magnitude of  $\mathbf{c}_{ij}$  is proportional to the topological asymmetry between  $v_i$  and  $v_j$ , as quantified by the aforementioned shortest path length difference (or other selected topological asymmetry metrics). This ensures that pairs with stronger topological asymmetry exert a larger rotational influence during the gradient update. The precise method for calculating  $\mathbf{c}_{ij}$  will be detailed in Section 5.

The weight  $w_{ij}$  associated with each chiral pair  $(v_i, v_j)$  reflects the relative importance of the chiral interaction. In this research,  $w_{ij}$  will initially be set to the reciprocal of the topological distance between  $v_i$  and  $v_j$ . This gives greater weight to topologically closer pairs, reflecting the observation in biological systems that closer neurons tend to have stronger interactions. We will also explore learning  $w_{ij}$  during training to allow for dynamic adaptation of the chiral influence based on data and learning progress. Alternative weighting schemes based on the degree of asymmetry or learned representations from a separate graph analysis will also be considered.

## 4 Mathematical Formulation

### 4.1 Definitions

- $\mathbf{c}_{ij}$ : The chiral vector for the pair of nodes  $(v_i, v_j)$ , representing the direction and magnitude of their chiral relationship. The precise method of calculating  $\mathbf{c}_{ij}$  based on topological features of the network is described in Section 5.
- $w_{ij}$ : A weight associated with the chiral pair  $(v_i, v_j)$ , reflecting the strength of their interaction. This might be a function of the topological distance or other properties of the chiral relationship.
- $s(w_{ij}, \mathbf{c}_{ij})$ : A sigmoid function that modulates the influence of the chiral term based on the weight  $w_{ij}$  and the magnitude of  $\mathbf{c}_{ij}$ . This ensures that nearby chiral pairs have a stronger influence than distant pairs, in line with the biological observation that the strength of neural connections diminishes with distance. The specific form of this function will be determined experimentally, potentially allowing it to be something more sophisticated or adapted for specific use cases.

The first core innovation of CGD lies in its gradient calculation which incorporates chiral vectors:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla L(\boldsymbol{\theta}_t) + \beta \sum_{i,j \in C} s(w_{ij}, \mathbf{c}_{ij}) (\nabla L(\boldsymbol{\theta}_t) \times \mathbf{c}_{ij}) \quad (1)$$

Where:

- $\boldsymbol{\theta}_t$  represents the parameter vector at iteration  $t$ .
- $\alpha$  denotes the learning rate.
- $\nabla L(\boldsymbol{\theta}_t)$  is the gradient of the loss function at iteration  $t$ .
- $\beta$  represents the chirality parameter, which modulates the influence of the chiral vectors.
- $C$  denotes the set of chiral pairs being considered during the update step, which may vary at each iteration depending on the method or constraints being used by the researcher.
- $w_{ij}$  represents a weight associated with the chiral pair  $(i, j)$ , and may reflect asymmetry measures related to properties of their chiral topologies.
- $\mathbf{c}_{ij}$  represents the chiral vector for the chiral pair  $(i, j)$ .
- $\times$  represents the cross-product.
- $s(w_{ij}, \mathbf{c}_{ij})$  is a function designed to blend the chiral vector's influence with considerations based on a weight  $w_{ij}$ .

## 5 Training Intuition and Higher Dimensions

Understanding chirality in higher dimensions can begin with visualizing simple cases (2D and 3D), followed by mathematical generalization to 4D, 5D, and beyond. Training involves visualization, representation with vectors and matrices, implementing simple transformations, and generalizing to higher dimensions.

Understanding the role of chirality in higher dimensions requires a shift in perspective, as traditional geometric intuitions may not directly apply. By visualizing lower-dimensional cases and gradually extending these insights to higher dimensions, researchers can develop a deeper intuition for the impact of chiral dynamics on learning. The use of vector and matrix representations allows for the implementation of simple transformations that can be generalized, providing a framework for exploring the effects of chirality across various dimensional spaces.

## 6 Towards CGD: A Synthesis

Chiral Gradient Descent (CGD) aims to enhance the efficiency and robustness of standard gradient descent by incorporating chirality—a concept of asymmetry—into the optimization process. This asymmetry is inspired by the prevalence of chiral structures and functions in biological systems, suggesting that introducing similar principles in optimization algorithms could lead to advantages not seen in more traditional, gradient-based methods.

### 6.1 The Chiral Update Rule

The second core innovation of CGD is its update rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla L(\boldsymbol{\theta}_t) + \beta \sum_{i,j \in C(\boldsymbol{\theta}_t)} \frac{|\mathbf{c}_{ij}|}{1 + e^{-\gamma d_{ij}}} (\nabla L(\boldsymbol{\theta}_t) \times \mathbf{c}_{ij}) \quad (2)$$

Where:

- $\boldsymbol{\theta}_t$ : Parameter vector at time  $t$ .
- $\alpha$ : Learning rate.
- $\nabla L(\boldsymbol{\theta}_t)$ : Gradient of the loss function.
- $\beta$ : Global chirality parameter.
- $C(\boldsymbol{\theta}_t)$ : Set of relevant chiral pairs, potentially changing dynamically during training.
- $\mathbf{c}_{ij}$ : Chiral vector for pair  $(i, j)$ , representing the direction and magnitude of the chiral influence (calculated using methods detailed in Section 5).

- $d_{ij}$ : Topological distance between nodes  $i$  and  $j$ , reflecting their structural relationship. This could be shortest-path distance, a measure of graph centrality, or other relevant metrics, depending on the properties of the data.
- $\gamma$ : Parameter controlling the sigmoid function’s steepness, determining the sensitivity to topological distance. Larger values of  $\gamma$  result in a sharper transition in the sigmoid function, while smaller values result in a more gradual transition, providing a mechanism to adjust the influence of distance on the weights.

## 6.2 Dynamic Chiral Pair Selection: $C(\theta_t)$

The set of relevant chiral pairs,  $C(\theta_t)$ , is dynamically determined at each iteration  $t$ . We prioritize pairs whose corresponding gradient magnitudes exceed a threshold  $\delta$ , and whose asymmetry scores (as defined in Section 5) are above a threshold  $\tau$ . This focuses computation on areas with active learning and significant asymmetry. Furthermore, we restrict the selection to pairs within a topological radius  $r$  to prevent long-range interactions from dominating the chiral update. For recurrent networks, this selection process will also incorporate temporal dependencies by prioritizing pairs with correlated activation patterns over a short time window. The specific values of  $\delta$ ,  $\tau$ , and  $r$  will be determined experimentally.

- **Gradient Magnitude:** Prioritize pairs whose corresponding gradients exceed a certain threshold, focusing on areas of the network where learning is most active.
- **Topological Distance:** Include pairs within a certain topological radius, preventing long-range chiral interactions from overwhelming the update.
- **Asymmetry Score:** Incorporate an asymmetry score (e.g., based on the cosine similarity between the feature embeddings of the chiral pair, as discussed in Section 5). Select pairs whose asymmetry scores exceed a specific threshold, focusing on the most significant asymmetries within the network structure.
- **Temporal Dynamics:** For recurrent networks, introduce a temporal component into the selection process, considering factors like previous activation patterns or temporal correlations.

## 6.3 Biological Inspiration

The sigmoid function in Equation 4 is inspired by the graded nature of synaptic weights in biological neural systems. The weight ( $w_{ij}$ ) can be interpreted as reflecting the strength of the chiral interaction, analogous to synaptic efficacy. The sigmoid function ensures that the chiral term’s influence decreases smoothly with increasing distance, mirroring how the influence of a neuron on its neighbors diminishes with physical distance in biological circuits.

## 7 Identifying Chiral Pairs: A Topologically-Informed Approach

This section details a novel method for identifying chiral pairs within complex networks, extending the approach described in Zhang *et al.* [2] for identifying topological invariants. Instead of directly predicting topological invariants, we adapt their convolutional neural network (CNN) architecture to identify pairs of nodes exhibiting chiral topological features, focusing on asymmetries within the network’s structure and information flow. This will form Phase 1 of our system for identifying chiral pairs to be used in subsequent phases to implement chiral gradient descent.

### 7.1 Network Representation

As in the previous sections, we represent networks as directed graphs  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of directed edges. Each edge  $e_{ij} \in E$  connecting node  $v_i$  to node  $v_j$  has an associated weight  $w_{ij}$  representing the strength of the connection (e.g., correlation between node activations, information flow, or interaction strength). We extend the graph representation by including additional node attributes that might influence the identification of chiral pairs, such as node centrality, community membership, and other topological measures that could prove useful in identifying pairs in the network. This extended representation is richer and more nuanced compared to simple directed graphs and is necessary to capture the more complex relationships between nodes in the network.

As in the previous sections, we represent networks as directed graphs  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of directed edges. Each edge  $e_{ij} \in E$  connecting node  $v_i$  to node  $v_j$ .

### 7.2 Adapting the Convolutional Neural Network

We adapt the CNN architecture proposed in Zhang *et al.* [2] (see Figure 1 in the original paper) to learn local topological features related to chirality. The input to the CNN will be a matrix representation of the local neighborhood around each node in the graph. This representation will be constructed by including several elements:

**Node Attributes:** Include node attributes such as centrality and community membership in the input matrix. **Edge Weights:** The edge weights from the node to its neighbors are added to the input matrix. **Shortest Path Lengths:** Compute the shortest path lengths between each pair of neighbors, which will inform the computation of the chiral vector in the subsequent phases.

This extended input representation captures both local topology and more global network features. This information is crucial for accurately identifying chiral pairs, unlike the approach in the original paper which only focused on computing the winding number.

### 7.3 Chiral Pair Identification

The output of the CNN is a vector that represents a topological feature embedding of the local neighborhood for each node. We define a chiral pair as a pair of nodes whose topological feature embeddings show a high degree of asymmetry or anti-correlation. This asymmetry or anti-correlation is evaluated using a distance metric, such as cosine similarity. We select the top pairs that maximize the asymmetry as the chiral pairs relevant to performing CGD in subsequent phases. The selection process could be made more complex to filter out pairs with specific traits or incorporate more data to reduce computational cost or improve performance.

## 8 Identifying Chiral Pairs: A Topologically-Informed Approach

This section details a method for identifying chiral pairs, building upon Zhang *et al.* [2] but incorporating novel elements to capture asymmetry relevant to narrative structures and social networks. This forms Phase 1 of our system, providing the foundation for chiral gradient descent.

### 8.1 Network Representation

We represent networks as directed graphs  $G = (V, E)$  with weighted edges  $w_{ij}$ . Crucially, we augment this with **relative path information**. For each node  $v_i$ , we pre-compute:

- **Shortest Path Lengths:** The shortest path length from  $v_i$  to every other node  $v_j$  (denoted  $d(v_i, v_j)$ ).
- **Common Ancestors and Path Differences:** For each pair of nodes  $(v_i, v_j)$ , we identify their common ancestors and calculate the *difference* in path lengths from each common ancestor to  $v_i$  and  $v_j$ . This captures the asymmetry in how  $v_i$  and  $v_j$  relate to their shared history or context.

This richer representation goes beyond simple connectivity, encoding structural asymmetries relevant to narratives (e.g., how different characters relate to key events) and social networks (e.g., differing perspectives on shared information).

### 8.2 Convolutional Neural Network

We adapt the CNN architecture from Zhang *et al.* [2] to learn chiral features. The input to the CNN for node  $v_i$  now includes:

- **Local Neighborhood Structure:** Edge weights  $w_{ij}$  for  $v_j$  in  $v_i$ 's neighborhood.
- **Relative Path Information:** For each neighbor  $v_j$ , include the shortest path length  $d(v_i, v_j)$  and the differences in path lengths from common ancestors.

This combined input allows the CNN to learn features sensitive to both local connectivity and global topological asymmetries.

### 8.3 Chiral Pair Identification

Instead of simply using cosine distance, we define a more nuanced chirality score:

$$\text{ChiralScore}(v_i, v_j) = \text{Asymmetry}(F_i, F_j) \times \text{PathDifference}(v_i, v_j) \quad (3)$$

Where:

- $F_i, F_j$  are the feature embeddings from the CNN for nodes  $v_i$  and  $v_j$ .
- $\text{Asymmetry}(F_i, F_j)$  measures the asymmetry between the embeddings (e.g., using cosine distance or a learned metric).
- $\text{PathDifference}(v_i, v_j)$  is a weighted average of the path length differences from common ancestors, emphasizing structural asymmetry.

Pairs with high ChiralScore are identified as chiral pairs. This combined score captures both feature-level and structural asymmetry.

### 8.4 Algorithm

---

#### Algorithm 1 Chiral Pair Identification

---

**Require:** Graph  $G = (V, E)$ , CNN model, chirality threshold  $\tau$

- 1: Pre-compute shortest path lengths and path differences for all node pairs
  - 2: Initialize empty set of chiral pairs  $C$
  - 3: **for** each node  $v_i \in V$  **do**
  - 4:     Construct input matrix  $M_i$  (neighborhood, paths)
  - 5:      $F_i \leftarrow \text{CNN}(M_i)$
  - 6: **end for**
  - 7: **for** each pair of nodes  $(v_i, v_j) \in V \times V$  **do**
  - 8:      $S_{ij} \leftarrow \text{ChiralScore}(v_i, v_j)$
  - 9:     **if**  $S_{ij} > \tau$  **then**
  - 10:          $C \leftarrow C \cup \{(v_i, v_j)\}$
  - 11:     **end if**
  - 12: **end for** **return**  $C$
-



## 8.5 Discussion

This method directly addresses the challenge of defining and detecting chirality relevant to narrative and social network analysis. By incorporating relative path information and a combined chirality score, it captures more nuanced asymmetries than simply comparing feature embeddings. The use of common ancestor paths adds a "historical" or "contextual" dimension to the chirality measure, which is particularly relevant for understanding how information and relationships evolve in narratives and social networks. The hierarchical application and the flexibility in defining the Asymmetry and PathDifference functions provide adaptability for various applications.

## 8.6 Hierarchical Application

To handle hierarchical structures, we apply the method recursively to sub-graphs within the network. This hierarchical application identifies chiral pairs at different scales, allowing for a more granular analysis of asymmetry in multi-level structures, which mirrors the multi-scale nature of hierarchical structures in narratives and social networks.

## 8.7 Algorithm

---

**Algorithm 2** Chiral Pair Identification

---

**Require:** Graph  $G = (V, E)$ , CNN model, asymmetry threshold  $\tau$

- 1: Initialize empty set of chiral pairs  $C$
  - 2: **for** each node  $v_i \in V$  **do**
  - 3:     Construct input matrix  $M_i$  for  $v_i$ 's neighborhood
  - 4:     Obtain feature embedding vector  $F_i$  from  $\text{CNN}(M_i)$
  - 5: **end for**
  - 6: **for** each pair of nodes  $(v_i, v_j) \in V \times V$  **do**
  - 7:     Compute asymmetry score  $A_{ij} = d(F_i, F_j)$  (e.g., using cosine distance)
  - 8:     **if**  $A_{ij} > \tau$  **then**
  - 9:         Add  $(v_i, v_j)$  to  $C$
  - 10:     **end if**
  - 11: **end for**
  - 12: **return** Set of chiral pairs  $C$
- 

## 8.8 Discussion

This method builds upon the success of CNNs in learning complex patterns from local data and provides a rigorous approach to identify chiral pairs. By incorporating topological features and hierarchical application, it addresses the complexities of analyzing asymmetries in real-world systems. The choice of CNN architecture and asymmetry threshold requires careful evaluation and tuning for specific applications and datasets.

We will compare this approach with alternative methods for measuring graph similarity and asymmetry in our validation experiments.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla L(\boldsymbol{\theta}_t) + \beta \sum_{i,j \in C(\boldsymbol{\theta}_t)} \frac{\|\mathbf{c}_{ij}\|}{1 + e^{-\gamma d_{ij}}} (\nabla L(\boldsymbol{\theta}_t) \times \mathbf{c}_{ij}) \quad (4)$$

Where:

- $\boldsymbol{\theta}_t$ : Parameter vector at time  $t$ .
- $\alpha$ : Learning rate.
- $\nabla L(\boldsymbol{\theta}_t)$ : Gradient of the loss function.
- $\beta$ : Global chirality parameter.
- $C(\boldsymbol{\theta}_t)$ : Set of relevant chiral pairs, potentially changing dynamically during training.
- $\mathbf{c}_{ij}$ : Chiral vector for pair  $(i, j)$ , calculated based on topology.
- $d_{ij}$ : Topological distance between  $i$  and  $j$  based on features like difference in path lengths, curvature measures, node or edge distribution densities, etc.
- $\gamma$ : Scaling parameter for the sigmoid function, influencing the impact of  $d_{ij}$ .

---

**Algorithm 3** Chiral Gradient Descent (CGD)

---

**Require:** Learning rate ( $\alpha$ ), chirality parameter ( $\beta$ ), scaling parameter ( $\gamma$ ), initial parameters ( $\boldsymbol{\theta}_0$ )

**while** not converged **do**

    Compute gradient:  $(\nabla L(\boldsymbol{\theta}_t))$

    Determine relevant chiral pairs:  $(C(\boldsymbol{\theta}_t))$  (using gradient magnitude, topological distance, asymmetry scores, and other factors)

    Calculate chiral vectors  $(\mathbf{c}_{ij})$  and distances  $(d_{ij})$  for  $((i, j) \in C(\boldsymbol{\theta}_t))$

$(\Delta \boldsymbol{\theta} = \beta \sum_{i,j \in C(\boldsymbol{\theta}_t)} \frac{\|\mathbf{c}_{ij}\|}{1 + e^{-\gamma d_{ij}}} (\nabla L(\boldsymbol{\theta}_t) \times \mathbf{c}_{ij}))$

    Update parameters:  $(\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla L(\boldsymbol{\theta}_t) + \Delta \boldsymbol{\theta})$

$(t \leftarrow t + 1)$

**end while**

---

This CGD algorithm uses a sigmoid function to modulate the influence of each chiral pair based on the topological distance  $d_{ij}$  between neurons. Chiral pairs that are topologically "closer" (smaller  $d_{ij}$ ) have a stronger influence on the update. The magnitude of the chiral vector  $\|\mathbf{c}_{ij}\|$  also contributes, allowing pairs with greater asymmetry to exert more influence. The parameter  $\beta$  controls the global effect of chirality, while  $\gamma$  modulates the sigmoid's steepness, providing control over the sensitivity to topological distances  $d_{ij}$ .

## 9 Methodology

This research will involve a phased approach, combining theoretical analysis, computational simulations, and experimental validation:

### 9.1 Phase 1: Chiral Pair Identification

We will implement the chiral pair identification method described in Section 5 using a Convolutional Neural Network (CNN). The CNN will be trained on synthetic graph datasets with varying topological properties, including networks generated using preferential attachment models and networks with known chiral structures. We will investigate different CNN architectures and hyperparameters to optimize the identification of chiral pairs. The performance will be evaluated using metrics like precision, recall, and F1-score, comparing our approach with baseline methods for graph similarity.

### 9.2 Phase 2: Chiral Gradient Descent Implementation

We will implement the CGD algorithm (Algorithm 3) in a deep learning framework (TensorFlow/PyTorch). Initial experiments will focus on simpler datasets (e.g., MNIST, CIFAR-10) and standard network architectures (e.g., Multilayer Perceptrons, Convolutional Neural Networks). We will explore different methods for calculating the chiral vectors  $\mathbf{c}_{ij}$ , weighting schemes for  $w_{ij}$ , and dynamic selection strategies for the chiral pair set  $C(\boldsymbol{\theta}_t)$ . The parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  will be tuned using grid search or Bayesian optimization.

### 9.3 Phase 3: Experimental Evaluation

We will evaluate CGD’s performance on more complex datasets (e.g., ImageNet) and larger network architectures (e.g., ResNet, Transformer). We will compare CGD with standard gradient descent methods (SGD, Adam) and other state-of-the-art optimizers. The evaluation metrics will include convergence speed, generalization performance (accuracy on a held-out test set), and robustness to noise and hyperparameter variations. Statistical significance testing (e.g., t-tests) will be used to compare the performance of different algorithms.

### 9.4 Datasets and Network Architectures

The research will utilize a variety of datasets, including:

- Standard image classification datasets (MNIST, CIFAR-10, ImageNet).

- Synthetic graph datasets with varying topological characteristics.
- Real-world social network datasets (if available and applicable).

The planned network architectures include:

- Multilayer Perceptrons (MLPs).
- Convolutional Neural Networks (CNNs).
- Graph Neural Networks (GNNs), if social network analysis is included.

## 9.5 Timeline and Milestones

- **Year 1:** Implement chiral pair identification (Phase 1) and CGD algorithm (Phase 2), preliminary tests on simple datasets.
- **Year 2:** Extensive experiments on complex datasets (Phase 3), compare CGD with baselines, refine the algorithm.
- **Year 3:** Apply CGD to novel architectures (e.g., GNNs), explore theoretical analysis of CGD’s convergence properties, disseminate findings.

## 10 Expected Outcomes and Discussion

We anticipate that CGD will outperform standard gradient descent methods, particularly in complex landscapes, by virtue of its enhanced exploration capabilities. We expect faster convergence and improved generalization performance, especially for datasets and network architectures that exhibit inherent chiral or asymmetric properties. We will analyze how different parameters and chiral vector calculation methods affect performance, aiming to identify the strengths and limitations of CGD. The open-source implementation of CGD will facilitate its adoption and further development by the research community, leading to novel approaches for optimization in various machine learning domains.

## 11 Conclusion

This research proposal presents a novel approach to gradient descent optimization that holds significant promise. By leveraging the power of chiral topologies and incorporating biologically plausible mechanisms into the optimization process, this research has the potential to overcome the limitations of traditional gradient descent and usher in a new era of more efficient and effective deep learning models. The research plan detailed above, if successfully executed, will provide valuable insight into the use of chiral gradient

descent and pave the way for its deployment in real-world applications. The next steps will involve developing and testing the CGD algorithm, conducting rigorous experiments, and analyzing the findings to validate its performance and contribute to the advancement of deep learning methodologies.

## References

- [1] Induraj. Different variants of gradient descent. <https://induraj2020.medium.com/different-variants-of-gradient-descent-6854d3118c4c>, 2023. Accessed: 2024-12-04.
- [2] P. Zhang, H. Shen, and H. Zhai. Machine learning topological invariants with neural networks. *Physical Review B*, 97(20):205138, 2018.