University of L'Aquila

Master Thesis

# SPN: A novel neural network architecture to improve the performance of MLPs

*Author:*
Sarosh Krishan

*Supervisor:*
Dr. Phuong T. Nguyen

*Corso di Laurea Magistrale in Informatica*

Department of Information Engineering, Information Sciences and Mathematics

Academic Year 2024/2025

**This thesis was conducted within the Erasmus Mundus Joint Master Degree Programme on the Engineering of Data-intensive Intelligent Software Systems (EDISS).**

# *Acknowledgements*

I would like to express my deep gratitude to my supervisor, Professor Phuong T. Nguyen, for his thoughtful advice, and support. His guidance has been indispensable throughout this process.

I would like to extend my thanks to EDISS programme coordinator Professor Sebastien Lafond and Professor Henry Muccini for providing me with this opportunity. Special thanks to Letizia Giorgetta for helping me navigate the intricacies of the administrative processes and always being ready to help.

I am incredibly blessed to have a wonderful support system of family and friends. I am extremely grateful to my parents Usha Bai and Krishan Lal, and my sisters Neev Kiran and Kiranti Krishan, for their love, unwavering support, and encouragement.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **LAH** | **L**ist **A**bbreviations **H**ere |
| **WSF** | **W**hat (it) **S**tands **F**or |

# Chapter 1

# Introduction

The Perceptron, introduced by Frank Rosenblatt in 1958, is one of the earliest and most influential models in machine learning (ML). Designed to solve binary classification problems, the perceptron is a simple linear classifier inspired by the biological neurons in the human brain. It works by assigning weights to a set of input features and calculating a weighted sum of these features, which is passed through an activation function to determine the classification. The model iteratively adjusts these weights based on prediction errors, learning from the data until it reaches an optimal configuration.

However, while the perceptron performs well for linearly separable data, it struggles with more complex, non-linear problems. This limitation prompted the development of the Multi-Layer Perceptron (MLP) in the 1980s. MLPs extend the perceptron by adding multiple layers of perceptrons, enabling the model to learn non-linear relationships within the data. The breakthrough of backpropagation, introduced by Paul Werbos in the 1970s and popularized by Geoffrey Hinton and others in the 1980s, made it possible to train MLPs by calculating gradients and adjusting the weights across multiple layers.

The evolution of neural networks has given rise to specialized architectures such as Convolutional Neural Networks (CNNs) for image tasks and Transformer-based Large Language Models (LLMs) for natural language processing. Despite this, the MLP remains a foundational architecture that laid the groundwork for more advanced models. MLPs consist of multiple layers where each neuron in one layer is fully connected to every neuron in the subsequent layer, enabling the model to learn complex patterns in data, such as those found in image and speech recognition.

Recent research in neural network architecture design has evolved into a field known as Neural Architecture Search (NAS), which aims to find optimal architectures for a specific task by exploring various combinations of hyperparameters, layers, and connections. NAS employs search strategies, such as reinforcement learning, evolutionary algorithms, and gradient-based methods, to efficiently navigate the vast space of possible architectures. Despite the progress made, traditional MLPs are constrained by hierarchical connections between layers, which limit their ability to explore more flexible connectivity patterns, such as those involving skip connections within or between layers. While advancements like Residual Networks (ResNet) have introduced skip connections to alleviate this issue, the full potential of neuron connectivity remains untapped.

This thesis proposes a novel approach, Sarosh's Perceptron Networks (SPNs), to address these limitations by enabling unrestricted connectivity between neurons. Unlike traditional MLPs, SPNs eliminate the constraints on layer-to-layer connections, allowing neurons to interact more freely across the network. By fostering greater flexibility in connectivity, SPNs are hypothesized to enhance model performance without introducing the computational overhead typically associated with large, fully connected networks. This work aims to improve the expressiveness and efficiency of neural network architectures, offering new avenues for both theoretical exploration and practical applications in machine learning.

## 1.1 Background and Motivation

The Perceptron marked the first significant step in the evolution of artificial neural networks (ANNs). Rosenblatt's model was inspired by the brain's structure, where neurons are connected by synapses and can adjust their strengths (synaptic weights) based on learning experiences. The Perceptron algorithm iterates over a set of input data, adjusts weights, and improves its ability to classify data. However, the perceptron could only solve problems that were linearly separable, leading to the need for more sophisticated architectures.

In the early 1980s, the Multi-Layer Perceptron (MLP) emerged, and the backpropagation algorithm was introduced to train these networks. Backpropagation allowed the MLPs to learn complex, non-linear functions by adjusting the weights across multiple layers using gradient descent. This innovation enabled the training of deep networks and formed the basis for modern deep learning techniques.

While MLPs were revolutionary, challenges remained regarding the optimal configuration of these networks. The number of layers and neurons in each layer, as well as how to connect these neurons, remained an open question. While the basic idea was that deeper networks (with more layers) could learn more complex patterns, no clear guidelines existed for how deep or wide the networks should be. Researchers began exploring empirical approaches to determine the ideal architecture.

One of the key insights came from "Efficient Backprop" (1998) by Yann LeCun, Léon Bottou, Geneviève Orr, and Klaus-Robert Müller, which highlighted methods for optimizing backpropagation. They proposed better weight initialization, momentum, and adaptive learning rates to accelerate convergence and prevent models from getting stuck in local minima. These methods made it possible to train deeper networks more effectively.

Further breakthroughs in architecture design came with Deep Residual Networks (ResNets), introduced in 2015 by Kaiming He and colleagues. ResNets utilized skip connections (residual blocks) that allowed information to bypass intermediate layers. This innovation solved the vanishing gradient problem, enabling the training of networks with hundreds or thousands of layers.

However, despite these advancements, the architectural design of neural networks remains an empirical challenge. Neural Architecture Search (NAS) emerged as an automated way to discover optimal architectures, but it is still computationally expensive and often depends on pre-existing knowledge.

## 1.2 Problem Statement

Although MLPs and other deep learning models have made significant strides in various tasks, the design of their architecture remains a critical issue. Traditional MLPs suffer from a limited connectivity between neurons. Neurons within a single layer do not interact with one another, and neurons in different layers are connected only via intermediate neurons. This design limits the network's ability to learn complex, interdependent features in the data.

Despite advancements such as skip connections and residual networks, these solutions do not fully address the limitations of traditional MLP architectures. There is still a gap in the literature regarding the optimal arrangement of weights and neuron connectivity, which could potentially improve model performance, reduce training time, and allow for smaller, more efficient networks.

## 1.3 Research Gap and Contributions

There is a significant gap in the literature concerning the optimization of neural network architectures, particularly in terms of neuron connectivity. Traditional MLPs rely on fixed architectural structures that restrict how neurons in different layers interact. Although skip connections have improved deep networks by mitigating the vanishing gradient problem, they still impose certain constraints, such as requiring neurons to have the same dimensionality or limiting connections to adjacent layers.

This thesis introduces Denser Perceptron Networks (DPNs), a new framework that eliminates the restriction of fixed connectivity patterns between neurons. DPNs allow for fully connected networks, where any two neurons can be linked. This increased connectivity has the potential to improve the model's ability to learn complex, interdependent features, potentially leading to improved generalization and performance.

The main contributions of this thesis are:

The introduction of DPNs: A new neural network architecture that eliminates the restrictions of traditional MLP designs.

Empirical evaluation: A comparison of DPNs with traditional MLPs to determine whether the increased connectivity results in better performance, faster training, and more efficient models.

Improved training efficiency: Investigation into whether DPNs can achieve high performance while reducing the time and memory costs typically associated with larger, deeper networks.

## 1.4 Research Questions

This thesis seeks to answer the following research questions:

Can Denser Perceptron Networks (DPNs) achieve improved model performance compared to traditional MLPs?

Does the removal of connectivity restrictions in neural networks improve the learning capabilities of the model?

Can DPNs maintain performance with smaller network sizes, thus improving computational efficiency?

How do DPNs compare to traditional architectures in terms of training time, memory consumption, and model accuracy?

How does the flexibility of DPNs affect their ability to generalize across different types of datasets and tasks?

## 1.5 Scope and Limitations

This research focuses on Denser Perceptron Networks (DPNs) and evaluates their potential to enhance neural network performance through improved connectivity. The scope of the thesis is limited to the empirical comparison of DPNs with traditional MLPs on classification tasks.

Key limitations include:

Computational Constraints: DPNs, by design, involve a large number of connections, which may result in higher computational demands for training and inference. This research will focus on evaluating whether the performance gains from DPNs justify the computational cost.

Task-specific Design: The results of this study may vary depending on the dataset and task. While the thesis focuses on classification tasks, the generalizability of DPNs to other types of tasks may require further investigation.

Data Constraints: The experiments will be conducted using a set of standard datasets, and the findings may not fully apply to more complex or diverse real-world data.

## 1.6 Thesis Structure and Overview

The remainder of this thesis is organized as follows:

1. Chapter 2 presents a literature review on the existing research on continual learning and the different methods proposed for the mitigation of catastrophic forgetting.
2. Chapter 3 describes the methodology used in the research, including the method for data preparation, preprocessing, fine-tuning, implementation of the mitigation approach, and the evaluation with different benchmarks.
3. Chapter 4 describes the different experiments carried out as part of this study, along with the different ablations performed.
4. Chapter 5 involves the analysis of the results obtained from the experiments.
5. Chapter 6 summarizes the findings of the thesis work and suggests directions for future work.

# Chapter 2

# Literature Review

## 2.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

### 2.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

### 2.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 2.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

# Chapter 3

# Methodology

## 3.1 Sarosh's Perceptron Networks (SPNs)

This thesis introduces Sarosh's Perceptron Networks (SPNs), a framework designed to eliminate restrictions on neuron connectivity. The goal is to allow any two neurons to connect, forming a directed acyclic graph (DAG) like network. This framework seeks to enhance neural networks by improving their connectivity while minimizing the associated increases in time and space complexities.

In theory, SPNs treat neurons as individual objects that can process inputs (also referred to as a forward pass) independently. These neurons may still depend on either the input data or other neurons for input, but the framework enables greater flexibility in forming connections across the network.

## 3.2 Maximal SPNs

To assess the time complexity of SPNs, we first examine the densest possible network configuration, known as Maximal SPNs. In this arrangement, neurons are connected in a sequential manner, where the first neuron is connected only to the input, the second neuron is connected to both the input and the first neuron, and so on. This structure leads to a fully connected network, where each neuron is linked to all preceding neurons, resulting in a Maximal SPN.

### 3.2.1 Object-based Representation with Partial Inputs and Propagation

One potential method to achieve the SPN framework is for neurons, as independent objects, to store partial inputs from sources that have completed their forward pass, while still waiting on outputs from other sources that haven't finished. Once a neuron receives all its necessary inputs, it performs its own forward pass and propagates its output to the subsequent neurons. However, this approach requires that the network be free of loops, as the presence of loops would result in deadlocks.

While this approach is conceptually simple and flexible, it is computationally expensive. Each neuron performs a storage and propagation step for every connection it has, leading to a large amount of redundancy. Even though the forward pass is only a single step once the inputs are complete, a worst-case scenario involves performing n forward passes, where n is the total number of neurons. Additionally, since each neuron stores its partial inputs, the outputs from a single neuron are duplicated for every neuron it propagates to. With each neuron also storing weights for every input, this method doubles the memory used compared to a traditional MLP network. Time Complexity: $O(n^2)$, where n is the total number of neurons. Space Complexity: $O(2d * n^2)$, where d is the feature size of the input data.

### 3.2.2 Sequential Representation with Compounding Input

In this approach, we describe the densest form of a SPN network using a weight matrix, where each row corresponds to a neuron, and the columns represent the input features and the outputs from all preceding neurons, except for the final one. This results in a lower triangular matrix in a staircase shape.

Such a network contains all possible connections for a given number of neurons. Therefore, any other network with the same number of neurons is a subnetwork of this complete network. To process this network, neurons are processed sequentially, starting from the first row. The output of a neuron is appended to its input, providing the input for the subsequent neuron. By doing so, we eliminate the need to add partial inputs for each neuron individually. Instead, the same input can be compounded across the forward pass.

This approach assumes a maximum connection for every neuron. Disconnecting two neurons is achieved by zeroing out the weight value for the output neuron corresponding to the input. This method resembles traditional MLP pruning, where weight values are zeroed out instead of being removed entirely.

While this method eliminates the propagation step from every neuron to its subsequent neurons, it still requires each neuron to temporarily store its input during the backpropagation process. Therefore, this approach does not reduce space complexity.

Time Complexity: $O(n)$.

Space Complexity: $O(2d * n^2)$.

### 3.2.3 Sequential Representation with Shared Input

This approach is identical to the previous one, with the key difference being that instead of storing inputs temporarily, neurons index slices from an input variable shared across the entire network. This indexing does not add significantly to the time complexity but optimizes it by eliminating redundancy.

Time Complexity: $O(n)$.

Space Complexity: $O(d * n^2)$.

### 3.2.4 Block-based Representation with Partial Outputs and Shared Input

When considering the lower triangular weight matrix and visualizing vertical lines at the edge of each neuron's weight vector, we see the formation of blocks within the matrix. The largest block contains the weights corresponding to the input features, and subsequent blocks contain the weights for the output of each neuron.

Rather than processing each neuron's weight vector individually, this approach processes the network one block at a time. The top-most neuron's forward pass is completed first, followed by calculating partial outputs for the remaining neurons. This method improves the time complexity by prioritizing the heaviest calculations (typically when the hardware is under lower load), and adds slight parallelization, as an input value is accessed only once, rather than repeatedly in a sequential approach.

This method is both energy and time-efficient, offering significant improvements over the previous approaches.

Time Complexity: $O(n)$.

Space Complexity: $O(d * n^2)$.

## 3.3 Free Weights

When projecting traditional MLPs onto a lower triangular matrix, we notice weight blocks isolated both vertically and horizontally. These isolated weights represent the independent layers in traditional MLPs. The right side of the matrix defines the time complexity of the forward pass; the more layers present, the longer the processing time. However, the left side of the matrix contains Free Weights, zeroed-out weights that do not contribute to time complexity.

Using free weights in traditional MLPs is similar to concatenating the output of a layer to its input before passing it to the next layer. This method increases the space complexity of the MLP to $O(n^2)$ while keeping the time complexity at O(l), where l is the number of layers. A major advantage of this approach is that subsequent layers can learn features based on both the input and the features from other previous layers, enabling the network to learn more complex patterns.

## 3.4 Minimal SPNs

While maximal SPNs maximize the number of connections in a perceptron network, they come with the trade-off of significantly increasing the time complexity from O(l) (where l is the number of layers) to O(n) (where n is the total number of neurons). Since l « n in most perceptron networks, this added time complexity becomes a significant challenge. On the other end of the spectrum, Minimal SPNs seek to minimize the number of blocks in the SPN framework. There are two possibilities:

1. If the total number of neurons is equal to the output size (n == o), only one block is needed, equivalent to a single MLP layer of size n.

2. If the total number of neurons exceeds the output size (n > o), the network must have at least two layers: one layer of size n - o for the input, and a second layer of size o that takes both the input and the output of the first layer.

Minimal SPNs provide the best time complexity of O(1) and the best space complexity of $O(n^2)$.

## 3.5 Maximal SPNs with Pruning

The final approach to improving SPN efficiency is pruning, inspired by the lottery ticket hypothesis. According to this hypothesis, there exists a smaller subnetwork within a neural network that can perform similarly to the original network if trained on the same dataset. The process involves training the parent network briefly (e.g., for one epoch), pruning the network slightly, and then resetting the remaining weights to their original values before repeating the first step for a desired number of iterations, until the network reaches a desired size. Then, the best performing pruned version of the network is trained for the complete duration.

This approach is applied to Maximal SPNs by zeroing out weights along the right edge of the lower triangular matrix. This pruning allows some neurons to share the same right-side edges, enabling parallel processing. The time complexity of the network is improved to O(l), where l is the number of vertical edges on the right side of the SPN matrix. Additionally, this method can reduce the space complexity by up to 90%, making it highly efficient.

# Chapter 4

# Experiments

## 4.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

### 4.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

### 4.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 4.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

# Chapter 5

# Results

## 5.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

### 5.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

### 5.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 5.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

# Chapter 6

# Conclusion

## 6.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

### 6.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

### 6.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

## 6.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

# Appendix A

# Frequently Asked Questions

## A.1 How do I change the colors of links?

The color of links can be changed to your liking using:
`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.
If you want to completely hide the links, you can use:
`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.
If you want to have obvious links in the PDF but not the printed text, use:
`\hypersetup{colorlinks=false}`.