



الجمهورية الجزائرية الديمقراطية الشعبية  
People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي  
Ministry of Higher Education and Scientific Research

جامعة غرداية  
University of Ghardaia

كلية العلوم والتكنولوجيا  
Faculty of Science and Technology

قسم الرياضيات والإعلام الآلي  
Department of Mathematics and Computer Science



Registration N°:  
...../...../...../...../.....

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

## Master

**Domain:** Mathematics and Computer Science

**Field:** Computer Science

**Specialty:** Intelligent Systems for Knowledge Extraction

## Theme

---

# Adaptive Traffic Signal Control Using Reinforcement Learning and Simulation

---

Presented by

*MEHAYA Fatima* & *ABDELHADI Imane*

**Publicly defended on June 21, 2026**

Jury members:

|                     |     |                |            |
|---------------------|-----|----------------|------------|
| Slimane OULADNAOUI  | MCA | Univ. Ghardaia | President  |
| Youcef MAHDJOUB     | MCA | Univ. Ghardaia | Examiner   |
| Messaoud BENGUENANE | MCB | Univ. Ghardaia | Supervisor |

**Academic Year: 2025 - 2026**

# *Acknowledgment*

*First and foremost, we would like to express our sincere gratitude to Almighty God for granting us the strength, patience, and perseverance to complete this work and overcome the challenges encountered throughout our academic journey.*

*We would like to extend our deepest appreciation and heartfelt thanks to our respected supervisor, Dr. BENGUENANE Messaoud , for his continuous support, valuable guidance, constructive feedback, and constant encouragement throughout the completion of this work.*

*We also sincerely thank all our esteemed professors who contributed to our academic training and supported us throughout our university studies.*

*In particular, we would like to acknowledge Dr. BELLAOUAR Slimane, Dr. OULADNAOUI Slimane, Dr. Mahdjoub Youcef, Dr. HASSINI Abdelmalek, Dr. Ben ABDELRAHMANE Habiba, and Dr. SAIDI Ahmed for their knowledge, guidance, and encouragement.*

*Finally, we would like to express our gratitude to everyone who, directly or indirectly, contributed to the completion of this work.*

## إهداء

الحمد لله ربّ العالمين، الذي أنعم عليّ بالقوة والتوفيق لإتمام هذا العمل وإخراجه إلى حيّز الوجود .

أتقدم بخالص الشكر وعظيم الامتنان إلى المشرف الدكتور بن قنان مسعود على توجيهاته القيّمة، ونصائحه السديدة، ودعمه المتواصل طوال جميع مراحل إنجاز هذا المشروع.

كما أتوجه بالشكر والتقدير إلى جميع أساتذة قسم الرياضيات والإعلام الآلي على ما قدموه لنا من علم ومعرفة وخبرة طوال سنوات الدراسة، وعلى جهودهم المخلصة في تكويننا الأكاديمي.

وأخصّ بالشكر عائلتي العزيزة على دعمها اللامحدود، وتشجيعها المستمر، وصبرها الدائم. لقد كانت مصدر قوتي الحقيقي طوال مسيرتي الجامعية، وأهديها هذا النجاح بكل فخر وامتنان.

كما أتقدم بجزيل الشكر إلى صديقتي رجاء وزهرة على دعمهما وتشجيعهما ومرافقتهما لي خلال فترة إنجاز هذا العمل.

وفي الختام، أتوجه بالشكر إلى كل من ساهم، بشكل مباشر أو غير مباشر، في إتمام هذا العمل، وساندني أو قدّم لي المساعدة بأي صورة كانت.

أسأل الله تعالى أن يتقبل هذا العمل وأن يجعله نافعا ومفيدا لكل من يطّلع عليه.

-----هـ مهابة فاطمة هـ-----

## إهداء

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
﴿قُلْ اَعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ﴾

ما سلكنا البدايات إلا بتيسيره، وما بلغنا النهايات إلا بتوفيقه، وما حققنا الغايات إلا بفضلِهِ،  
سبحانه وتعالى .

إلى من بلغ الرسالة، وأدى الأمانة، ونصح الأمة، إلى نبي الرحمة ونور العالمين، سيدنا محمد صلى  
الله عليه وسلم.

إلى نفسي التي بدأت بطموح وانتهت بنجاح، إلى تلك الروح التي حملت في قلبها يقيناً بأن معية الله  
تكفي، أهديك هذا الإنجاز، فكنت الأولى في مسيرة الصبر والمثابرة.

إلى الذي زين اسمي بأجمل الألقاب، من حصد الأشواك ليمهد لي درب العلم، أبي الغالي.  
إلى من كان دعاؤها سر نجاحي، ومن علمتني أن الإرادة تصنع المعجزات، أمي الغالية.  
إلى تلك النجوم التي تثير طريقي دوماً، صنّاع قوتي وصفوة أيامي، الذين كانوا شركاء كل خطوة،  
أختي وأخي، سلمى وعبد الرحمن.

إلى جدي وجدتي، اللذين لم يبخلا عليّ بالدعاء، أسياد القلب وأغلى أحبابه، حفظكم الله وأطال  
عمركما.

إلى رفيقة الدرب في البدايات والنهايات، في العثرات والنجاحات، صديقة قلبي لينة.  
إلى معلمي الأفاضل، الذين حملوا على عاتقهم رسالة الأنبياء، وسقونا من معين العلم، وغرسوا فينا  
القيم قبل المعارف.

وفي الختام، عربون حب وامتنان لكل من شاركني الطريق وترك بصمة في رحلتي، إلى من آمنوا  
بي حين شككت في نفسي، ورفعوا رأسي حين أثقلني التعب.  
شكراً لكم جميعاً.

﴿وما توفيقي إلا بالله عليه توكلت وإليه أنيب﴾

# ملخص

يُعدّ الازدحام المروري الحضري من أبرز التحديات التي تواجه المدن الحديثة، حيث يكشف عن القيود الأساسية لأنظمة التحكم التقليدية في إشارات المرور المعتمدة على التوقيت الثابت. تهدف هذه الأطروحة إلى اقتراح وتقييم إطار عمل للتعلم المعزز يراعي العدالة في التحكم التكيفي بإشارات المرور، وذلك بالاعتماد على خوارزمية تحسين السياسة القريبة Proximal Policy Optimisation (ppo) وتنفيذه داخل محاكي المرور المجهرى SUMO.

تم تصميم دالة مكافأة متعددة الأهداف بهدف تحقيق التوازن بين كفاءة حركة المرور والعدالة بين الاتجاهات المرورية، وذلك من خلال دمج عدة معايير تشمل تقليل التأخير، تقليل أطوال الطوابير، زيادة معدل مرور المركبات وتعظيم مؤشر العدالة Jain's Fairness Index.

وقد تم تقييم ثلاث استراتيجيات مختلفة لتوزيع الأوزان داخل دالة المكافأة: استراتيجية موجهة نحو الكفاءة واستراتيجية موجهة نحو العدالة واستراتيجية متوازنة تجمع بينهما. كما تمت مقارنة خمسة متحكمات بصورة منهجية، وهي: Fixed-Time, Vanilla PPO, PPO Efficiency, PPO Fairness, PPO Balanced وذلك عبر سيناريوهين مرورين مختلفين: مسار تدريب مألوف (Training Route) مع طلب ثابت غير متوازن ومسار غير مرئي (Unseen Route) مع طلب متماثل متغير زمنياً لتقييم التعميم بدون تدريب إضافي (zero-shot).

تظهر النتائج التجريبية على مسار التدريب أن جميع المتحكمات القائمة على PPO تقلل متوسط زمن الانتظار بنسبة 40-47% مقارنة بخط الأساس ذو التوقيت الثابت (Fixed-Time). يحقق PPO Efficiency أعلى إنتاجية تدريبية بين متحكمات التعلم المعزز (655 مركبة) وأقل زمن انتظار (15.63 ثانية).

يقدم PPO Balanced الأداء متعدد الأهداف الأكثر اتساقاً، محققاً أقل متوسط طول لقائمة الانتظار (168.2 مركبة).

على المسار غير المرئي يظهر PPO Fairness تعميماً متفوقاً محتفظاً بنسبة 130.7% من إنتاجيته التدريبية (459 ← 600 مركبة) مع الحفاظ على مؤشر Jain للعدالة بقيمة 0.8314 وهو أعلى بكثير من عتبة القبول 0.8.

ومن الجدير بالذكر أن جميع المتحكمات الموجهة بالعدالة تحافظ على عدالة اتجاهية مقبولة تحت تأثير التحول في التوزيع (distribution shift).

ومن أهم النتائج المستخلصة أن تضمين مفهوم العدالة داخل دالة المكافأة يؤدي بشكل غير مباشر إلى تنظيم السياسة المتعلمة وتحسين متانتها عند مواجهة تغيرات في توزيع البيانات أو الظروف المرورية غير المألوفة وتوفر هذه النتائج إرشادات عملية يمكن الاستفادة منها عند نشر أنظمة التحكم الذكية في إشارات المرور ضمن تطبيقات المدن الذكية.

---

الكلمات المفتاحية: التحكم التكيفي في إشارات المرور، التعلم المعزز، تحسين السياسة القريبة (PPO)، العدالة، مؤشر العدالة Jain، SUMO، دالة مكافأة متعددة الأهداف، التعميم، المدينة الذكية.

# Abstract

Urban traffic congestion remains a critical challenge for modern cities, exposing the fundamental limitations of conventional fixed-time signal controllers. This thesis proposes and evaluates a fairness-aware reinforcement learning (RL) framework for adaptive traffic signal control, built around the Proximal Policy Optimisation (PPO) algorithm and implemented within the SUMO microscopic traffic simulator.

A multi-objective reward function is designed to jointly optimise traffic efficiency and directional fairness, combining delay reduction, queue minimisation, throughput maximisation, and Jain's Fairness Index. Three distinct weight configurations are evaluated: efficiency-oriented, fairness-oriented, and balanced. Five controllers are systematically compared: Fixed-Time, Vanilla PPO, PPO Efficiency, PPO Fairness, and PPO Balanced across two traffic scenarios: a familiar Training Route (with static imbalanced demand) and an Unseen Route (with temporally-varying symmetric demand) for zero-shot generalisation assessment.

Experimental results on the Training Route demonstrate that all PPO-based controllers reduce average waiting time by 40–47% relative to the Fixed-Time baseline. PPO Efficiency achieves the highest training throughput among RL controllers (655 vehicles) and the lowest waiting time (15.63 s). PPO Balanced provides the most consistent multi-objective performance, achieving the lowest average queue length (168.2 vehicles).

On the Unseen Route, PPO Fairness demonstrates superior generalisation, retaining 130.7% of its training throughput (459 → 600 vehicles) while maintaining a Jain's Index of 0.8314, well above the 0.8 acceptability threshold. Notably, all fairness-weighted controllers maintain acceptable directional equity under distribution shift.

A key finding is that fairness-oriented reward shaping implicitly regularises the learned policy, improving robustness under distribution shift. These results offer practical deployment guidelines for smart city traffic management systems.

**Keywords:** Adaptive Traffic Signal Control · Reinforcement Learning · Proximal Policy Optimisation · Fairness · Jain's Fairness Index · SUMO · Multi-objective Reward · Generalisation · Smart City

# Résumé

La congestion du trafic urbain reste un défi critique pour les villes modernes, mettant en évidence les limites fondamentales des contrôleurs de feux de signalisation conventionnels à temps fixe. Cette thèse propose et évalue un cadre d'apprentissage par renforcement soucieux de l'équité pour le contrôle adaptatif des feux de signalisation, fondé sur l'algorithme d'optimisation des politiques proximales (PPO) et implémenté dans le simulateur de trafic microscopique SUMO.

Une fonction de récompense multi-objectif est conçue pour optimiser conjointement l'efficacité du trafic et l'équité directionnelle, combinant la réduction des retards, la minimisation des files d'attente, la maximisation du débit et l'indice d'équité de Jain. Trois configurations de poids distinctes sont évaluées : orientée efficacité, orientée équité et équilibrée. Cinq contrôleurs sont systématiquement comparés Fixed-Time, Vanilla PPO, PPO Efficiency, PPO Fairness et PPO Balanced à travers deux scénarios de trafic : un itinéraire d'entraînement familier (Training Route, avec une demande statique déséquilibrée) et un itinéraire non vu (Unseen Route, avec une demande symétrique variable dans le temps) pour une évaluation de la généralisation zero-shot. Les résultats expérimentaux sur le Training Route démontrent que tous les contrôleurs basés sur PPO réduisent le temps d'attente moyen de 40 à 47 % par rapport à la ligne de base Fixed-Time. PPO Efficiency atteint le débit d'entraînement le plus élevé parmi les contrôleurs d'apprentissage par renforcement (655 véhicules) et le temps d'attente le plus bas (15,63 s). PPO Balanced offre les performances multi-objectifs les plus cohérentes, obtenant la plus faible longueur moyenne des files d'attente (168,2 véhicules).

Sur l'Unseen Route, PPO Fairness démontre une généralisation supérieure, conservant 130,7 % de son débit d'entraînement (459 → 600 véhicules) tout en maintenant un indice de Jain de 0,8314, bien au-dessus du seuil d'acceptabilité de 0,8. Notamment, tous les contrôleurs pondérés par l'équité maintiennent une équité directionnelle acceptable sous l'effet du décalage de distribution.

Un résultat clé est que la mise en forme de la récompense orientée vers l'équité régularise implicitement la politique apprise, améliorant ainsi la robustesse face au décalage de distribution. Ces résultats offrent des lignes directrices pratiques pour le déploiement de systèmes de gestion du trafic dans les villes intelligentes.

**Mots-clés :** Contrôle adaptatif des feux de circulation · Apprentissage par renforcement · Proximal Policy Optimization (PPO) · Équité · Indice d'équité de Jain · SUMO · Fonction de récompense multi-objectifs · Généralisation · Ville intelligente.

# CONTENTS

|                                                                   |            |
|-------------------------------------------------------------------|------------|
| <b>List of Figures</b>                                            | <b>vi</b>  |
| <b>List of Tables</b>                                             | <b>vii</b> |
| <b>Introduction</b>                                               | <b>1</b>   |
| <b>1 Background</b>                                               | <b>3</b>   |
| 1.1 Introduction . . . . .                                        | 3          |
| 1.2 Urban Traffic Congestion and Traffic Signal Control . . . . . | 3          |
| 1.3 Intelligent Transportation Systems and Smart Cities . . . . . | 4          |
| 1.4 Reinforcement Learning for Traffic Signal Control . . . . .   | 4          |
| 1.5 Traffic Simulation Environment . . . . .                      | 5          |
| 1.6 Fairness in Traffic Signal Control . . . . .                  | 5          |
| <b>2 Literature Review</b>                                        | <b>6</b>   |
| 2.1 Introduction . . . . .                                        | 6          |
| 2.2 Overview of Adaptive Traffic Control Techniques . . . . .     | 6          |
| 2.2.1 Fixed-Time Control . . . . .                                | 6          |
| 2.2.2 Actuated Control . . . . .                                  | 7          |
| 2.2.3 Coordinated Adaptive Systems . . . . .                      | 7          |
| 2.2.4 Modern AI-Based Adaptive Control . . . . .                  | 7          |
| 2.3 Reinforcement Learning Foundations . . . . .                  | 8          |
| 2.3.1 Basic Concept . . . . .                                     | 8          |
| 2.3.2 Value-Based Methods . . . . .                               | 8          |
| 2.3.3 Policy-Based Methods . . . . .                              | 8          |
| 2.3.4 Actor-Critic Methods . . . . .                              | 9          |
| 2.3.5 Multi-Agent Reinforcement Learning . . . . .                | 9          |
| 2.4 RL-Based Traffic Signal Control Studies . . . . .             | 10         |
| 2.4.1 Single-Intersection Control . . . . .                       | 10         |

---

|          |                                                |           |
|----------|------------------------------------------------|-----------|
| 2.4.2    | Multi-Intersection Control                     | 10        |
| 2.4.3    | State Representation                           | 11        |
| 2.4.4    | Reward Design                                  | 11        |
| 2.4.5    | Action Space Design                            | 11        |
| 2.5      | Fairness in Traffic Control                    | 11        |
| 2.5.1    | Definition of Fairness                         | 11        |
| 2.5.2    | Importance of Fairness                         | 11        |
| 2.5.3    | Fairness Metrics                               | 12        |
| 2.5.4    | Existing Fairness-Aware Methods                | 12        |
| 2.6      | Comparison of Simulation Tools                 | 12        |
| 2.6.1    | SUMO                                           | 12        |
| 2.6.2    | CityFlow                                       | 12        |
| 2.6.3    | PTV VISSIM                                     | 13        |
| 2.6.4    | Other Tools                                    | 13        |
| 2.6.5    | Simulation Tools Comparison                    | 13        |
| 2.6.6    | Tool Selection                                 | 13        |
| 2.7      | Existing Limitations and Research Gaps         | 14        |
| 2.7.1    | Limitations of Traditional Systems             | 14        |
| 2.7.2    | Limitations of RL-Based Systems                | 14        |
| 2.7.3    | Efficiency–Fairness Trade-off                  | 14        |
| 2.7.4    | Research Gaps Summary                          | 14        |
| 2.7.5    | Thesis Contribution                            | 15        |
| 2.8      | Algorithm Selection and Simulation Environment | 16        |
| 2.9      | Summary                                        | 16        |
| <b>3</b> | <b>System Architecture and Methodology</b>     | <b>17</b> |
| 3.1      | Overview of the Proposed RL Framework          | 17        |
| 3.2      | SUMO Traffic Simulation Env                    | 18        |
| 3.2.1    | Description of the Simulation Environment      | 18        |
| 3.2.2    | Urban Road Network Design                      | 18        |
| 3.2.3    | Traffic Demand and Route Configuration         | 19        |
| 3.2.4    | Unseen Route Evaluation                        | 19        |
| 3.2.5    | Simulation Duration and Training Robustness    | 19        |
| 3.3      | Experimental Environment Config                | 20        |
| 3.3.1    | SUMO Installation and Configuration            | 20        |
| 3.3.2    | TraCI Configuration                            | 21        |
| 3.3.3    | Python Environment and Required Libraries      | 21        |
| 3.3.4    | Verification of the Complete Stack             | 22        |
| 3.4      | IM-SASD                                        | 23        |

---

|        |                                                          |    |
|--------|----------------------------------------------------------|----|
| 3.4.1  | State Space . . . . .                                    | 23 |
| 3.4.2  | Action Space . . . . .                                   | 24 |
| 3.4.3  | Control Objectives . . . . .                             | 24 |
| 3.5    | Reward Function Formulation . . . . .                    | 25 |
| 3.5.1  | Delay Component . . . . .                                | 26 |
| 3.5.2  | Queue Length Component . . . . .                         | 26 |
| 3.5.3  | Throughput Component . . . . .                           | 26 |
| 3.5.4  | Fairness Component (Jain’s Fairness Index) . . . . .     | 27 |
| 3.5.5  | Imbalance Penalty . . . . .                              | 27 |
| 3.5.6  | Weight Configurations . . . . .                          | 27 |
| 3.6    | RL Algorithm Choice . . . . .                            | 28 |
| 3.6.1  | Overview of Proximal Policy Optimisation (PPO) . . . . . | 28 |
| 3.6.2  | The PPO Objective Function . . . . .                     | 29 |
| 3.6.3  | Advantages of PPO for Traffic Signal Control . . . . .   | 29 |
| 3.7    | RL Model Implementation . . . . .                        | 30 |
| 3.7.1  | Framework Implementation . . . . .                       | 30 |
| 3.7.2  | Neural Network Architecture . . . . .                    | 30 |
| 3.7.3  | Logging and Checkpoint System . . . . .                  | 31 |
| 3.7.4  | Evaluation Strategy . . . . .                            | 32 |
| 3.8    | TP-EC . . . . .                                          | 33 |
| 3.8.1  | Training Procedure . . . . .                             | 33 |
| 3.8.2  | Hyperparameter Settings . . . . .                        | 34 |
| 3.8.3  | Multi-Seed Training . . . . .                            | 35 |
| 3.8.4  | Pareto-Based Experiments . . . . .                       | 36 |
| 3.8.5  | Best Model Selection . . . . .                           | 36 |
| 3.9    | Baseline Implementation . . . . .                        | 37 |
| 3.9.1  | Fixed-Time Controller . . . . .                          | 37 |
| 3.9.2  | Vanilla PPO . . . . .                                    | 37 |
| 3.9.3  | Comparison Methodology . . . . .                         | 37 |
| 3.10   | Experimental Scenarios . . . . .                         | 38 |
| 3.10.1 | Imbalanced Traffic Scenario . . . . .                    | 38 |
| 3.10.2 | Fairness-Oriented Experiments . . . . .                  | 38 |
| 3.10.3 | Efficiency-Oriented Experiments . . . . .                | 38 |
| 3.10.4 | Unseen Route Evaluation . . . . .                        | 39 |
| 3.11   | IF-RLAS . . . . .                                        | 40 |
| 3.11.1 | Integration Architecture . . . . .                       | 40 |
| 3.11.2 | Simulation Initialisation . . . . .                      | 41 |
| 3.11.3 | State Observation Collection . . . . .                   | 41 |
| 3.11.4 | RL Decision-Making Process . . . . .                     | 42 |

---

|          |                                                                      |           |
|----------|----------------------------------------------------------------------|-----------|
| 3.11.5   | Traffic Signal Execution . . . . .                                   | 42        |
| 3.11.6   | Reward Computation and Learning Feedback . . . . .                   | 43        |
| 3.11.7   | Continuous Closed-Loop Interaction . . . . .                         | 43        |
| 3.11.8   | Real-Time Communication Benefits . . . . .                           | 44        |
| 3.12     | Chapter Summary . . . . .                                            | 44        |
| <b>4</b> | <b>Results and Analysis</b>                                          | <b>46</b> |
| 4.1      | Introduction . . . . .                                               | 46        |
| 4.2      | Experimental Evaluation Setup . . . . .                              | 46        |
| 4.2.1    | Simulation Environment . . . . .                                     | 46        |
| 4.2.2    | Evaluated Controllers . . . . .                                      | 47        |
| 4.2.3    | Evaluation Metrics . . . . .                                         | 48        |
| 4.2.4    | Evaluation Scenarios . . . . .                                       | 49        |
| 4.3      | Comparative Performance on Training Route . . . . .                  | 49        |
| 4.3.1    | Training Route Performance Summary . . . . .                         | 49        |
| 4.3.2    | Average Waiting Time . . . . .                                       | 50        |
| 4.3.3    | Total Throughput . . . . .                                           | 51        |
| 4.3.4    | Fairness Jain's Index . . . . .                                      | 52        |
| 4.3.5    | Average Travel Time . . . . .                                        | 53        |
| 4.3.6    | Average Queue Length . . . . .                                       | 54        |
| 4.3.7    | Episode Reward Comparison . . . . .                                  | 56        |
| 4.3.8    | Directional Fairness: EW vs NS Waiting Time . . . . .                | 57        |
| 4.3.9    | Overall Performance Profile Radar Chart . . . . .                    | 59        |
| 4.4      | Generalisation Performance on Unseen Route . . . . .                 | 60        |
| 4.4.1    | Waiting Time on Unseen Route . . . . .                               | 62        |
| 4.4.2    | Throughput Generalisation . . . . .                                  | 64        |
| 4.4.3    | Fairness Generalisation . . . . .                                    | 65        |
| 4.5      | Discussion of Results . . . . .                                      | 67        |
| 4.5.1    | Efficiency-Fairness Trade-Off Analysis . . . . .                     | 67        |
| 4.5.2    | Why Fairness-Oriented Agents Generalize Better . . . . .             | 68        |
| 4.5.3    | Overfitting Evidence in PPO Efficiency . . . . .                     | 69        |
| 4.5.4    | Correlation Between Reward and Real Traffic Metrics . . . . .        | 70        |
| 4.5.5    | The Fixed-Time Baseline Unexpected Unseen-Route Throughput . . . . . | 70        |
| 4.5.6    | Practical Deployment Recommendations . . . . .                       | 71        |
| 4.5.7    | Summary of Key Findings . . . . .                                    | 72        |
| 4.6      | Limitations of the Study . . . . .                                   | 73        |
| 4.7      | Summary . . . . .                                                    | 73        |
| <b>5</b> | <b>Conclusion and Future Work</b>                                    | <b>75</b> |
| 5.1      | Conclusion . . . . .                                                 | 75        |

---

|                           |    |
|---------------------------|----|
| 5.2 Future Work . . . . . | 76 |
|---------------------------|----|

# LIST OF FIGURES

|      |                                                                                                                                                                                              |    |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1  | Reinforcement Learning Interaction Loop for Adaptive Traffic Signal Control [33]                                                                                                             | 17 |
| 3.2  | Simulated 2×2 Urban Road Network in SUMO                                                                                                                                                     | 18 |
| 3.3  | SUMO (Simulation of Urban MObility) Traffic Simulation Platform                                                                                                                              | 20 |
| 3.4  | MDP Formulation for RL-Based Traffic Signal Control                                                                                                                                          | 24 |
| 3.5  | PPO Actor–Critic Neural Network Architecture [34]                                                                                                                                            | 31 |
| 3.6  | Decision-making loop of the PPO-based traffic signal controller.                                                                                                                             | 34 |
| 3.7  | Integration Architecture Between SUMO, TraCI, and Python [35]                                                                                                                                | 40 |
| 3.8  | Interaction sequence between the training script, PPO agent, Gymnasium environment, TraCI, and SUMO simulator                                                                                | 41 |
| 4.1  | presents the average waiting time for all five controllers on the Training Route.                                                                                                            | 50 |
| 4.2  | shows total throughput (vehicles per episode) on the Training Route.                                                                                                                         | 51 |
| 4.3  | illustrates Jain’s Fairness Index for controllers configured with explicit fairness weights ( <i>PPO Fairness</i> , <i>PPO Balanced</i> , and <i>PPO Efficiency</i> ) on the Training Route. | 52 |
| 4.4  | Average travel time results on the Training Route.                                                                                                                                           | 53 |
| 4.5  | presents the average queue length across the network on the Training Route.                                                                                                                  | 54 |
| 4.6  | presents the mean episode reward achieved by each controller on the Training Route.                                                                                                          | 56 |
| 4.7  | disaggregates waiting time by traffic direction (EW vs NS) under imbalanced demand conditions.                                                                                               | 57 |
| 4.8  | presents a normalized radar chart consolidating all five performance metrics across controllers.                                                                                             | 59 |
| 4.9  | presents a side-by-side comparison of waiting times on both routes.                                                                                                                          | 62 |
| 4.10 | presents the absolute throughput change (Unseen TP - Training TP).                                                                                                                           | 64 |
| 4.11 | shows side-by-side Jain’s Index values on Training and Unseen Routes.                                                                                                                        | 65 |
| 4.12 | presents the absolute change (Unseen - Training).                                                                                                                                            | 66 |

# LIST OF TABLES

|     |                                                                           |    |
|-----|---------------------------------------------------------------------------|----|
| 2.1 | Comparison of Traffic Signal Control Approaches . . . . .                 | 8  |
| 2.2 | Comparison of Reinforcement Learning Methods in Traffic Control . . . . . | 10 |
| 2.3 | Comparison of Simulation Tools for Traffic Signal Control . . . . .       | 13 |
| 3.1 | Required Python libraries and their roles . . . . .                       | 22 |
| 3.2 | Experimental Environment and Configuration . . . . .                      | 23 |
| 3.3 | Reward weight configurations for the evaluated models . . . . .           | 28 |
| 3.4 | PPO Hyperparameter Configuration . . . . .                                | 35 |
| 3.5 | Traffic observation metrics and corresponding TraCI functions . . . . .   | 42 |
| 4.1 | Training Route Performance Summary . . . . .                              | 49 |
| 4.2 | EW vs NS Directional Waiting Time Imbalanced Traffic Scenario . . . . .   | 58 |
| 4.3 | Generalisation Performance Summary Training vs Unseen Route . . . . .     | 61 |
| 4.4 | Waiting Time: Training vs Unseen Route Comparison . . . . .               | 62 |
| 4.5 | Total Throughput (TP): Training vs Unseen Route Comparison . . . . .      | 64 |
| 4.6 | Jain’s Fairness Index: Training vs Unseen Route Comparison . . . . .      | 66 |
| 4.7 | Practical Controller Deployment Recommendations . . . . .                 | 71 |

# Introduction

On a daily basis, millions of drivers sit at red lights watching time slip away not because the roads are too narrow, but because the signals controlling them are not smart enough. Traffic congestion has become one of the most persistent and costly challenges facing modern cities. Beyond the frustration of daily commuters, it has real consequences: wasted fuel, rising emissions, delayed emergency responses, and an urban quality of life that quietly erodes year after year. As cities grow faster than their infrastructure can adapt, the limitations of conventional traffic management have never been more visible.

For decades, traffic signals have operated on fixed schedules green for thirty seconds, red for thirty seconds regardless of whether ten cars are waiting or a hundred. This approach was practical when traffic was predictable and sparse. Today, it is increasingly inadequate. Urban traffic is dynamic, asymmetric, and sensitive to events that no pre-programmed timer can anticipate. Something smarter is needed.

Reinforcement learning offers a compelling answer. Unlike traditional rule-based systems, an RL agent learns directly from experience: it observes the traffic state, chooses a signal action, receives feedback, and gradually builds a policy that responds intelligently to real conditions. Research over the past decade has shown that RL-based controllers can substantially outperform fixed-time baselines in terms of waiting time, queue length, and throughput. Yet a persistent blind spot remains in most of these approaches: they optimize for average efficiency while ignoring whether that efficiency is distributed fairly across all traffic directions. In practice, this often means that minor roads and low-volume approaches suffer disproportionately long delays so that the dominant traffic stream moves faster a trade-off that raises legitimate questions about equitable service in public infrastructure.

This thesis addresses that gap directly. It proposes and rigorously evaluates a fairness-aware reinforcement learning framework for adaptive traffic signal control, built around the Proximal Policy Optimisation algorithm. The framework is implemented and evaluated within the SUMO microscopic traffic simulator. The central question driving this work is not simply whether RL can outperform fixed-time control that much is already well established but how the design of the reward function shapes the relationship between traffic efficiency

---

and directional fairness, and what that means for real-world deployment. Five controllers are compared across two traffic scenarios, including an unseen route designed to test how well learned policies transfer to unfamiliar demand patterns. The results reveal a richer picture than efficiency metrics alone would suggest: fairness-oriented reward shaping not only improves equity but also produces policies that generalize more robustly under distribution shift, while purely efficiency-focused agents show signs of over-fitting to their training environment. These findings carry practical implications for how smart city traffic systems should be designed, evaluated, and deployed.

The thesis is organized as follows:

**Chapter 1** introduces the research context, motivation, and objectives.

**Chapter 2** reviews the relevant literature on adaptive traffic signal control, reinforcement learning methods, and fairness in transportation systems.

**Chapter 3** describes the proposed framework in detail, covering the simulation environment, reward function design, algorithm configuration, and experimental setup.

**Chapter 4** presents the results and analysis across all performance metrics and evaluation scenarios.

**Chapter 5** concludes with a synthesis of the key findings, practical deployment recommendations, and directions for future research.

# Chapter 1

## Background

### 1.1 Introduction

Urban traffic congestion has become one of the most pressing challenges in modern cities due to rapid urbanization and the continuous growth in vehicle ownership [1, 2]. Efficient traffic signal control is therefore essential for improving traffic flow, reducing delays, and minimizing environmental and economic impacts [4, 14]. While conventional traffic signal controllers have been widely adopted, their limited ability to adapt to dynamic traffic conditions has motivated the development of intelligent control strategies [14, 29, 30]. Recent advances in Artificial Intelligence, particularly Reinforcement Learning (RL), have enabled the design of adaptive traffic signal controllers capable of learning optimal decision-making policies through interaction with the traffic environment [12, 15, 17]. This chapter presents the theoretical background and fundamental concepts underlying this work, including traffic signal control, Intelligent Transportation Systems, Reinforcement Learning, the Markov Decision Process, traffic simulation, and fairness-aware traffic management.

### 1.2 Urban Traffic Congestion and Traffic Signal Control

Traffic congestion occurs when the demand for road space exceeds the available capacity, resulting in reduced vehicle speeds, longer travel times, and increased queue lengths. It is influenced by several factors, including high traffic demand, road infrastructure limitations, traffic incidents, and inefficient intersection management [1, 2, 4]. Since signalized intersections are the primary points where different traffic streams interact, their operation has a significant impact on the overall performance of urban transportation networks [6, 8, 14].

Traffic signal control regulates vehicle movements by allocating the right of way through a sequence of green, yellow, and red phases [14, 16]. According to their operating principles, traffic signal controllers are commonly classified into fixed-time, actuated, and adaptive systems [14, 29, 30]. Fixed-time controllers rely on predefined timing plans, whereas actuated

controllers adjust signal timings based on sensor information. However, both approaches have limited adaptability to continuously changing traffic conditions [14, 29, 30]. To overcome these limitations, Adaptive Traffic Signal Control (ATSC) dynamically modifies signal timings according to real-time traffic conditions, enabling more efficient intersection management and improved traffic flow [6, 17].

### 1.3 Intelligent Transportation Systems and Smart Cities

Intelligent Transportation Systems (ITS) integrate sensing technologies, communication networks, and intelligent control techniques to improve the efficiency, safety, and sustainability of transportation systems [5]. By collecting and processing real-time traffic data, ITS enables adaptive traffic management and more effective use of existing road infrastructure. As a key component of Smart Cities, ITS supports the development of intelligent mobility solutions capable of responding to dynamic traffic conditions [5, 13]. Recent advances in Artificial Intelligence have further strengthened ITS by enabling learning-based traffic signal control methods that continuously optimize traffic flow through real-time decision making [12, 13].

### 1.4 Reinforcement Learning for Traffic Signal Control

Reinforcement Learning (RL) is a branch of Machine Learning in which an agent learns how to make decisions through continuous interaction with an environment. At each step, the agent observes the current state of the environment, selects an action, and receives a reward that evaluates the quality of its decision. Through repeated interactions, the agent gradually learns a policy that maximizes long-term cumulative rewards [15]. In the context of traffic signal control, the agent corresponds to the traffic controller, the environment represents the traffic network, the state describes current traffic conditions such as queue lengths and waiting times, and the actions correspond to changing or maintaining traffic signal phases [6, 15, 17].

Most reinforcement learning problems are modeled as a "Markov Decision Process (MDP)", commonly represented as  $(MDP = (S, A, P, R, \gamma))$ , where (S) denotes the set of states, (A) the set of possible actions, (P) the state transition probability, (R) the reward function, and ( $\gamma$ ) the discount factor [15]. In traffic signal control, this formulation enables the controller to learn how its actions influence future traffic conditions and to optimize signal timings accordingly [15, 17]. To address complex traffic environments, "Deep Reinforcement Learning (DRL)" combines RL with neural networks, allowing the agent to process high-dimensional traffic information [11, 19]. Among the various DRL algorithms, "Proximal Policy Optimization (PPO)" has gained significant attention due to its stable learning process and strong performance in adaptive traffic signal control applications [20, 22, 26].

## 1.5 Traffic Simulation Environment

Developing and evaluating intelligent traffic signal controllers directly on real road networks is often impractical due to safety, cost, and operational constraints. Therefore, microscopic traffic simulation provides a safe and flexible environment for testing and validating control strategies under various traffic scenarios. In this work, the "Simulation of Urban MObility (SUMO)" is used to model traffic networks and vehicle movements, while "TraCI (Traffic Control Interface)" enables real-time communication between the simulator and the reinforcement learning agent [25]. To facilitate the training process, "Gymnasium" provides a standardized reinforcement learning environment [37], whereas "Stable-Baselines3" supplies efficient implementations of modern deep reinforcement learning algorithms, including Proximal Policy Optimization (PPO) [20]. Together, these tools form an integrated framework for developing, training, and evaluating the proposed adaptive traffic signal control system.

## 1.6 Fairness in Traffic Signal Control

The primary objective of most traffic signal control systems is to improve traffic efficiency by minimizing vehicle delays, reducing queue lengths, and increasing traffic throughput [10, 11]. However, optimizing these performance measures alone may lead to unfair traffic distribution, particularly under imbalanced traffic demand, where major roads receive excessive green time while vehicles on minor approaches experience prolonged waiting times. This phenomenon, commonly referred to as the "starvation problem", reduces the quality of service for less congested directions [10, 18, 22]. Consequently, recent research has emphasized the importance of incorporating fairness into traffic signal control strategies, aiming to achieve a better balance between overall traffic efficiency and equitable service among different traffic approaches [10, 21, 22]. This balance between efficiency and fairness constitutes the main motivation behind the proposed approach presented in this thesis.

# Chapter 2

## Literature Review

### 2.1 Introduction

This chapter presents a comprehensive review of adaptive traffic signal control systems, with a focus on reinforcement learning-based approaches and fairness-aware traffic management. It outlines the evolution from conventional methods to intelligent systems, highlighting key contributions and limitations. The review also focuses on reinforcement learning as the main framework used in modern traffic signal control. In addition, special attention is given to fairness, which is a key objective in this thesis.

### 2.2 Overview of Adaptive Traffic Control Techniques

Traffic signal control systems have evolved significantly over time due to increasing urban traffic complexity. Different approaches have been proposed to improve efficiency and reduce congestion in road networks.

#### 2.2.1 Fixed-Time Control

Fixed-time control is the earliest method used in traffic signal systems. It relies on pre-defined timing plans based on historical traffic data. These plans remain constant regardless of real-time traffic conditions and are usually designed using methods such as Webster's formula [16]. The main advantage of fixed-time control is its simplicity and ease of implementation. It is also cost-effective and performs reasonably well under stable traffic conditions. However, this method lacks adaptability to dynamic traffic changes. It often becomes inefficient in real-world scenarios where traffic demand varies significantly during peak hours and unexpected events [17].

### 2.2.2 Actuated Control

Actuated control improves upon fixed-time systems by introducing real-time detection. Sensors such as inductive loops are used to detect vehicle presence and adjust signal timing accordingly [17]. This approach allows green time extension when vehicles are detected and reduces unnecessary waiting time when traffic demand is low. As a result, it provides better responsiveness compared to fixed-time control [17]. Despite this improvement, actuated systems remain limited in scope. They rely on local detection points and cannot capture the global traffic state. Therefore, their decision-making is reactive rather than predictive [12].

### 2.2.3 Coordinated Adaptive Systems

Coordinated systems such as SCATS and SCOOT were developed to manage traffic at a network level. These systems continuously adjust signal parameters based on real-time traffic measurements [12]. SCATS modifies cycle length, phase splits, and offsets dynamically, while SCOOT optimises signal timing by minimising delays and stops using upstream traffic data. These systems provide better coordination across intersections and have been successfully deployed in many cities. However, they are expensive to install and maintain, which limits their use in developing regions [17]. In addition, their underlying models are based on simplified traffic assumptions, which may not fully represent complex real-world traffic dynamics [12].

### 2.2.4 Modern AI-Based Adaptive Control

Recent advances in artificial intelligence have introduced reinforcement learning as a new approach for traffic signal control. Unlike traditional methods, RL-based systems learn optimal policies directly from interaction with the environment [12]. This approach removes the need for explicit traffic modelling and allows continuous improvement over time. It also adapts automatically to changing traffic patterns without manual tuning [15]. However, RL-based systems require large amounts of training data and careful design of state, action, and reward functions to perform effectively [12].

Various traffic signal control approaches have been proposed to improve traffic flow, ranging from conventional fixed-time methods to advanced reinforcement learning-based techniques. These approaches differ in their adaptability to changing traffic conditions, implementation complexity, deployment cost, scalability, and overall performance. Understanding these differences is essential for selecting the most appropriate control strategy for a given traffic environment. Table 2.1 presents a comparison of the main traffic signal control approaches based on these key characteristics

Table 2.1: Comparison of Traffic Signal Control Approaches

| Approach           | Adaptability | Complexity | Cost          | Performance | Scalability | Key Limitation                                                    |
|--------------------|--------------|------------|---------------|-------------|-------------|-------------------------------------------------------------------|
| Fixed-Time Control | Very Low     | Low        | Low           | Low         | Moderate    | No real-time adaptation [16]                                      |
| Actuated Control   | Moderate     | Moderate   | Low           | Moderate    | Moderate    | Limited to local decisions [14]                                   |
| SCATS / SCOOT      | High         | High       | Very High     | High        | High        | Expensive infrastructure and deployment requirements [29, 30]     |
| RL-Based Control   | Very High    | High       | Moderate-High | Very High   | Very High   | Requires extensive training data and computational resources [12] |

## 2.3 Reinforcement Learning Foundations

### 2.3.1 Basic Concept

Reinforcement Learning is a learning paradigm where an agent interacts with an environment to maximise cumulative reward. The process is modelled as a Markov Decision Process (MDP) [15]. The MDP is defined by the tuple  $(S, A, P, R, \gamma)$ . In traffic control, the state represents traffic conditions, actions represent signal decisions, and rewards measure system performance such as delay or queue length [12].

### 2.3.2 Value-Based Methods

Value-based methods aim to learn the value of state-action pairs. Q-learning is the fundamental algorithm in this category. However, it becomes inefficient in large-scale problems due to high dimensionality. Deep Q-Networks (DQN) solve this issue by using neural networks to approximate Q-values. Several improvements such as Double DQN and Duelling DQN improve stability and performance [15]. Rainbow combines multiple enhancements and has shown strong results in traffic signal control tasks [12].

### 2.3.3 Policy-Based Methods

Policy-based methods directly optimise the policy instead of estimating value functions. This makes them suitable for complex and continuous action spaces [15]. Proximal Policy

Optimisation (PPO) is one of the most widely used algorithms in this category. It ensures stable learning by limiting large policy updates using a clipped objective function [20]. This method has been successfully applied in traffic signal control systems with strong performance [12].

### 2.3.4 Actor-Critic Methods

Actor-critic methods combine both value-based and policy-based learning. The actor selects actions while the critic evaluates them. This structure improves learning stability and convergence speed [15]. Algorithms such as A2C and A3C have been widely used in traffic control applications [12].

### 2.3.5 Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning (MARL) is used when multiple agents interact in the same environment. In traffic systems, each intersection can be controlled by a separate agent [12]. However, MARL introduces challenges such as non-stationary environments and coordination difficulties. These challenges make learning more complex and less stable. To address this, frameworks like CTDE (Centralised Training, Decentralised Execution) are used. Methods such as VDN and QMIX help coordinate multiple agents effectively [12].

Reinforcement learning algorithms used in traffic signal control can be broadly classified into several categories, each with different characteristics, advantages, and limitations. Their suitability depends on the complexity of the traffic environment, the size of the road network, and the type of decision-making required. Table 2.2 summarises the main reinforcement learning method types, together with their strengths, weaknesses, and typical traffic control applications

Table 2.2: Comparison of Reinforcement Learning Methods in Traffic Control

| Method Type    | Examples        | Strengths                              | Weaknesses                               | Traffic Application Suitability |
|----------------|-----------------|----------------------------------------|------------------------------------------|---------------------------------|
| Value-Based    | Q-learning, DQN | Simple, stable for discrete actions    | Struggles with large state spaces        | Single intersections [15, 11]   |
| Policy-Based   | PPO             | Stable learning, continuous control    | Sensitive to hyperparameter tuning       | Complex traffic scenarios [20]  |
| Actor-Critic   | A2C, A3C        | Balanced learning, faster convergence  | Higher computational complexity          | Medium-scale networks [20]      |
| Multi-Agent RL | QMIX, VDN, CTDE | Scalable to multi-intersection systems | Coordination and non-stationarity issues | Large traffic networks [27, 18] |

## 2.4 RL-Based Traffic Signal Control Studies

### 2.4.1 Single-Intersection Control

Early studies showed that reinforcement learning can effectively control traffic signals at isolated intersections. Q-learning was first used successfully in this context [15]. Later, deep reinforcement learning improved scalability and performance. For example, DQN-based systems significantly reduced delay and improved efficiency [12]. Other works such as Li et al. achieved a 14% reduction in delay [19], while encoding-based state representations have been shown to significantly improve learning efficiency. Recent studies have shown that PPO-based reinforcement learning approaches can improve traffic efficiency and reduce queue lengths compared to traditional fixed-time and actuated signal control methods, depending on the simulation environment and traffic scenario [12].

### 2.4.2 Multi-Intersection Control

Multi-intersection control introduces coordination challenges between agents [12]. Early work used centralised approaches such as n-step SARSA to reduce complexity [9]. More recent studies use multi-agent reinforcement learning to improve coordination across networks. For example, MA-PPO achieved a 24% improvement in travel time [27]. Communication-based approaches and advanced architectures such as MTLIGHT further improve convergence speed and scalability [26].

### 2.4.3 State Representation

State representation plays a critical role in RL performance. Different methods have been proposed to represent traffic conditions. Some approaches use queue length vectors, while others use grid-based or image-based representations [12]. More advanced methods include encoded representations such as auto encoders . These methods improve learning efficiency but increase system complexity. Therefore, a trade-off exists between accuracy and computational cost [12].

### 2.4.4 Reward Design

Reward design directly influences agent behaviour. Common reward functions aim to minimise delay, reduce queue length, or maximise throughput . More advanced approaches use pressure-based metrics or composite reward functions that combine multiple objectives [12]. While a few recent studies have initiated attempts to incorporate fairness considerations into multi-intersection environments [22], these approaches often treat equity as a secondary metric or rely on localised thresholds. Comprehensive frameworks that rigorously optimise the structural trade-off between network-wide efficiency and directional equity under distribution shifts remain scarce, which represents a key focus of this work.

### 2.4.5 Action Space Design

Action space defines how the agent interacts with traffic signals. Common strategies include selecting phases, adjusting durations, or following predefined cycles . Flexible action spaces provide better control but increase learning complexity. Therefore, designing an appropriate action space is essential for system performance [12].

## 2.5 Fairness in Traffic Control

### 2.5.1 Definition of Fairness

Fairness in traffic control refers to ensuring equal treatment of different traffic flows. It aims to avoid situations where certain roads experience excessive waiting times compared to others . Two main types of fairness are commonly considered: delay-based fairness and throughput-based fairness [22].

### 2.5.2 Importance of Fairness

Fairness is important because it directly affects safety and user satisfaction. When certain roads are consistently disadvantaged, drivers may violate traffic rules, increasing accident

risks. It also improves public acceptance of intelligent traffic systems by ensuring equitable treatment across all users [22].

### 2.5.3 Fairness Metrics

Fairness can be measured using several indicators. Common metrics include maximum waiting time, variance of delays, and Gini coefficient. Another important metric is the equity factor  $\eta$ , which balances efficiency and fairness in reward design [22].

### 2.5.4 Existing Fairness-Aware Methods

Recent studies have started integrating fairness into reinforcement learning models. Yan et al. introduced a fairness-aware RL model that balances efficiency and equity [22]. FELight uses a threshold-based mechanism to activate fairness control only when needed, improving performance while maintaining fairness [21]. More recent work such as M2SAC extends fairness to multi-modal traffic environments, including vehicles, buses, and pedestrians [18].

## 2.6 Comparison of Simulation Tools

Simulation tools are essential for evaluating reinforcement learning-based traffic signal control systems. They provide realistic environments to test and validate proposed models.

### 2.6.1 SUMO

SUMO is one of the most widely used open-source traffic simulators. It supports microscopic traffic modelling and provides detailed vehicle-level simulation [25]. It is highly compatible with RL frameworks through the TraCI interface, making it suitable for real-time interaction with learning agents [25]. However, SUMO can become computationally expensive when simulating large-scale networks. This may limit training speed in reinforcement learning experiments [12].

### 2.6.2 CityFlow

CityFlow is designed specifically for large-scale multi-agent reinforcement learning applications. It provides faster simulation compared to SUMO and is optimized for urban traffic scenarios. It supports Python-based APIs, which makes integration with RL frameworks easier and more efficient [28]. Despite its speed advantage, CityFlow offers fewer modelling details compared to SUMO, which may reduce realism in some cases [12].

### 2.6.3 PTV VISSIM

VISSIM is a commercial high-fidelity traffic simulator widely used in industry and research . It provides very realistic traffic behaviour modelling and supports advanced simulation features . It is often used for validating RL models in realistic environments and supports integration with real traffic systems. However, its main limitation is its high cost and computational requirements, which restrict its use in academic environments [12].

### 2.6.4 Other Tools

Other simulation platforms include AIMSUN, MATSim, and Paramics. These tools offer different levels of accuracy and scalability depending on the application . Each simulator has its own trade-off between realism, speed, and accessibility, making tool selection dependent on research objectives [12].

### 2.6.5 Simulation Tools Comparison

Table 2.3: Comparison of Simulation Tools for Traffic Signal Control

| Simulator | Type        | Speed      | Realism   | Scalability | Cost      | RL Suitability           |
|-----------|-------------|------------|-----------|-------------|-----------|--------------------------|
| SUMO      | Open-source | Medium     | High      | Medium      | Free      | Very High [25]           |
| CityFlow  | Open-source | Very High  | Medium    | Very High   | Free      | Very High [28]           |
| VISSIM    | Commercial  | Low–Medium | Very High | High        | Expensive | High (industry use) [31] |
| AIMSUN    | Commercial  | Medium     | High      | High        | Expensive | Medium–High [32]         |

### 2.6.6 Tool Selection

For this research, SUMO is chosen as the primary simulation environment. This decision is supported by multiple considerations , as summarised in Table 2.3 . SUMO is the most frequently used simulation platform in RL-based traffic signal control studies due to its open-source nature, modular architecture, and real-time Python API (TraCI), which allows seamless integration with RL frameworks [25]. In contrast, CityFlow another popular open-source simulator is known for its high simulation speed (more than twenty times faster than SUMO) and strong scalability for large-scale, multi-agent scenarios [28]. However, given that the focus of this work is on microscopic traffic dynamics and the realism of individual vehicle behaviour rather than city-wide simulation speed, SUMO’s fine-grained traffic modelling is

better aligned with the research objectives. Therefore, SUMO is adopted as the simulation backbone, while CityFlow is acknowledged as a viable alternative for studies prioritising computational scalability.

## 2.7 Existing Limitations and Research Gaps

### 2.7.1 Limitations of Traditional Systems

Traditional traffic control systems such as fixed-time and actuated methods suffer from limited adaptability. They rely on predefined rules that cannot handle dynamic traffic variations [17]. Even coordinated systems like SCATS and SCOOT improve performance but remain expensive and difficult to deploy at large scale [12].

### 2.7.2 Limitations of RL-Based Systems

Despite their success, reinforcement learning-based traffic systems still face several challenges. Many approaches use simplified state representations that fail to capture full spatial traffic dynamics [12]. This reduces model accuracy in complex environments. In multi-intersection settings, scalability becomes a major issue due to exponential growth in state-action space [9]. Additionally, coordination between agents remains difficult due to non-stationary learning environments [12].

### 2.7.3 Efficiency–Fairness Trade-off

Most existing RL-based traffic systems focus mainly on efficiency metrics such as delay reduction or throughput maximisation. However, this often leads to unfair treatment of low-volume traffic flows, where some directions experience excessive waiting times. This creates a critical gap between efficiency and fairness, which has not been fully addressed in existing research [22].

### 2.7.4 Research Gaps Summary

Based on the literature review, the main research gaps are:

- **Weak spatial state representation:** Existing RL-based traffic signal control systems use simplified state representations (e.g., queue length vectors) that fail to capture full spatial traffic dynamics, which reduces model accuracy in complex environments [12].
- **Lack of scalable multi-intersection RL architectures:** In multi-intersection settings, scalability becomes a major issue due to exponential growth in state-action space. Co-

ordination between agents remains difficult due to non-stationary learning environments [12, 27].

- **Limited integration of fairness into reward design:** Most existing RL-based traffic systems focus mainly on efficiency metrics such as delay reduction or throughput maximisation, often at the expense of equitable treatment across different traffic flows [17, 21, 22].
- **Absence of unified efficiency–fairness optimisation frameworks:** While recent works such as FELight [21] and M2SAC [18] have made initial progress, a unified framework that simultaneously optimises both efficiency and fairness is still absent [22].
- **Limited real-world validation of RL-based systems:** Despite promising simulation results, most studies are conducted in simulated environments, and the transferability of learned policies to real-world traffic conditions has not been thoroughly investigated [12].

### 2.7.5 Thesis Contribution

This thesis addresses the identified research gaps by proposing a fairness-aware multi-intersection reinforcement learning system based on the Proximal Policy Optimisation (PPO) algorithm [20].

The main contributions of this work are threefold.

**First**, a fairness-integrated reward function is designed by incorporating Jain’s Fairness Index and an imbalance penalty to establish a flexible, multi-objective optimisation framework. Unlike existing approaches which mainly optimise static delay reduction or absolute throughput maximisation, the proposed reward function enables parametric tuning to explicitly balance competing efficiency and equity objectives. Specifically, the framework introduces Pareto-optimised configurations such as PPO Balanced and PPO Efficiency . These configurations demonstrate that introducing moderate fairness constraints implicitly acts as a policy regulariser ; it prevents side-street and low-volume traffic starvation while allowing the agent to dynamically prioritise high-volume corridors, thereby effectively reducing excessive waiting times for minor flows while maintaining a high overall traffic throughput across the network.

**Second**, the proposed framework extends RL-based control to multi-intersection environments through a centralised single-agent architecture with a comprehensive state representation that fuses real-time queue lengths, directional waiting times, imbalance metrics, and current phase status to capture essential spatial traffic dynamics.

**Third**, the system is evaluated using a dual set of metrics: efficiency metrics (average waiting time, queue length, throughput, travel time) and fairness metrics (Jain’s Fairness Index, dir-

ectional waiting time imbalance), ensuring a balanced and rigorous performance assessment across both objectives [22].

Collectively, these contributions establish a scalable and fairness-aware reinforcement learning framework capable of balancing operational efficiency with equitable traffic distribution in multi-intersection urban environments, directly addressing the limitations identified in Sections 2.7.1 to 2.7.4.

## 2.8 Algorithm Selection and Simulation Environment

Based on a review of previous studies in adaptive traffic signal control, several reinforcement learning approaches have been proposed, each with different strengths and limitations depending on traffic complexity and network scale. Experimental results in the literature also indicate that performance is highly sensitive to environment configuration, state representation, and traffic demand patterns.

Accordingly, the Proximal Policy Optimisation (PPO) algorithm was selected as the most suitable approach for this study due to its stable training behaviour, robustness in complex environments, and relatively simple implementation compared to other reinforcement learning methods [20]. PPO has been widely adopted in traffic signal control problems due to its ability to maintain stable policy updates through clipped objective functions.

This work adopts a centralised single-agent reinforcement learning framework to control multiple intersections within the network. The proposed system is implemented and evaluated using the SUMO (Simulation of Urban Mobility) environment [25] to ensure a realistic and controllable traffic simulation setting.

## 2.9 Summary

This chapter reviewed the evolution of traffic signal control systems from traditional methods to reinforcement learning-based approaches. It also examined RL foundations, state representations, reward designs, and simulation tools. Special attention was given to fairness-aware traffic control, which remains an under explored research area. The identified gaps motivate the methodology proposed in the next chapter.

# Chapter 3

## System Architecture and Methodology

### 3.1 Overview of the Proposed RL Framework

The proposed reinforcement learning framework integrates the Simulation of Urban Mobility (SUMO) microscopic traffic simulator with a deep reinforcement learning agent through the Traffic Control Interface (TraCI) real-time communication protocol [25]. This integration enables continuous closed-loop interaction in which the simulator provides traffic observations to the RL agent, while the RL agent returns adaptive traffic signal control actions to the simulator [6]. The framework follows a modular architecture consisting of three main components: **First**, the SUMO traffic simulation environment responsible for modelling vehicle movements, traffic flows, and intersection operations [25]; **Second**, the custom Gymnasium environment [37] that implements the state space, action space, and multi-objective reward function; and **Third**, the Proximal Policy Optimisation (PPO) agent that learns adaptive traffic signal control policies through repeated interaction with the environment [20].

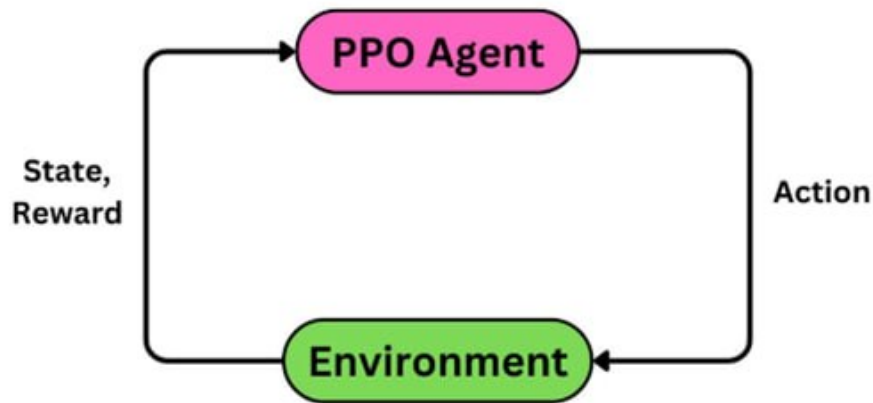


Figure 3.1: Reinforcement Learning Interaction Loop for Adaptive Traffic Signal Control [33]

## 3.2 SUMO-Based Traffic Simulation Environment

### 3.2.1 Description of the Simulation Environment

The traffic signal control environment was implemented using the Simulation of Urban MObility (SUMO) traffic simulator integrated with Python through the TraCI interface. SUMO is an open source microscopic traffic simulator developed by the German Aerospace Center (DLR) [25]. It was selected because it provides deterministic and reproducible traffic simulation, detailed vehicle-level control, and real-time communication support for reinforcement learning applications [25, 6]. The simulator was used to model vehicle movements, traffic flows, queue dynamics, and traffic signal operations within the urban road network. The microscopic simulation approach enables tracking of individual vehicle behaviours including acceleration, deceleration, lane changes, and intersection adherence, which is essential for accurate traffic signal control evaluation [25].

### 3.2.2 Urban Road Network Design

A custom 2×2 urban road network was created using the SUMO network file: 2x2.net.xml

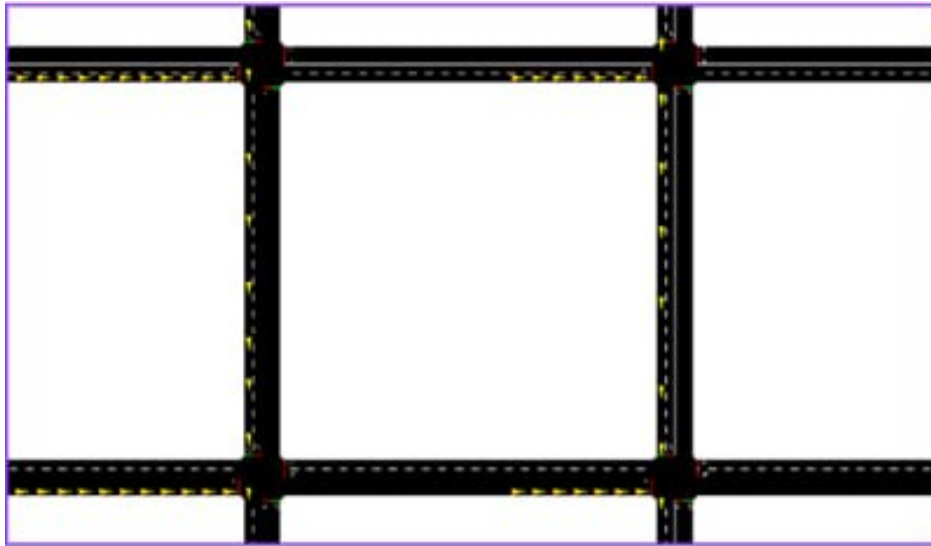


Figure 3.2: Simulated 2×2 Urban Road Network in SUMO

The Figure 3.2 shows the custom 2×2 urban traffic network implemented in SUMO, including multiple signalised intersections, lane connections, and bidirectional traffic flows used for reinforcement learning experiments. The network file defines:

- Road topology including all four approach directions
- Signalised intersections with complete lane geometry

- Lane connections for straight and turning movements
- Traffic signal configurations and phase definitions
- Vehicle movement directions and right of way rules

The simulated network represents a signalised urban traffic intersections with multiple incoming and outgoing traffic flows, including both straight and turning vehicle movements. Each approach direction contains multiple lanes to accommodate varying traffic volumes and turning movements. The network design enabled the evaluation of adaptive traffic signal control strategies under different traffic conditions including imbalanced flows, and congestion scenarios.

### 3.2.3 Traffic Demand and Route Configuration

Vehicle traffic demand was generated using SUMO route configuration files: `.rou.xml`. The route files define:

- Vehicle generation rates and inter arrival distributions
- Vehicle departure times and arrival patterns
- Traffic flow distributions across different directions
- Vehicle routing behaviour including origin destination pairs
- Vehicle type definitions including desired speed and acceleration parameters

Different traffic demand patterns were used during the experiments to simulate varying traffic conditions and congestion levels. Imbalanced traffic scenarios were considered to evaluate the behaviour of the reinforcement learning controller under different traffic distributions.

### 3.2.4 Unseen Route Evaluation

To evaluate the generalisation capability of the trained reinforcement learning agents, additional unseen traffic routes were introduced during testing. These unseen route scenarios were not used during the training phase, meaning the agent had no prior experience with these specific traffic patterns [12]. The objective of this evaluation was to determine whether the learned traffic signal control policies could adapt successfully to previously unseen traffic patterns and traffic distributions. This testing methodology helped assess the robustness, adaptability, and transferability of the proposed RL based traffic signal controller under unfamiliar traffic conditions without additional retraining [8, 12].

### 3.2.5 Simulation Duration and Training Robustness

Each simulation episode lasted 5400 simulation steps (corresponding to 5400 seconds or 90 minutes of simulated traffic time when using the 5-second control interval). Multiple random seeds were used during training to reduce over-fitting and improve policy stability across different traffic conditions. Using different seeds allowed the reinforcement learning

agent to experience diverse traffic dynamics during training and improved the reproducibility of experimental results by reducing the probability of obtaining biased results caused by favourable random initialisation conditions[12].

### 3.3 Experimental Environment Configuration

To ensure reproducibility and stable execution of the experiments, all software components were installed and configured to operate together correctly. This section documents the complete software stack used throughout the experimentation phase.

#### 3.3.1 SUMO Installation and Configuration

SUMO version 1.18.0 or later was used during the experiments. The installation process on Microsoft Windows included the following steps:

- Download and install the 64-bit SUMO package
- Use the default installation path: C:\Program Files (x86)\Eclipse\Sumo\
- Add the SUMO binary directory to the system PATH variable
- Create the environment variable SUMO-HOME

To verify the installation, the following command was executed: `sumo --version` Successful execution confirmed that SUMO was correctly installed and accessible from the command line.



Figure 3.3: SUMO (Simulation of Urban MObility) Traffic Simulation Platform

### 3.3.2 TraCI Configuration

The Traffic Control Interface (TraCI) was used to establish real time communication between Python and the SUMO simulator. Through TraCI, the reinforcement learning agent was able to observe traffic conditions and dynamically control traffic signal phases during simulation. The SUMO tools directory was added to the Python search path using:

```
import os, sys

if 'SUMO_HOME' not in os.environ:
    os.environ['SUMO_HOME'] = r'C:\Program Files (x86)\Eclipse\
        Sumo '

sys.path.append(os.path.join(os.environ['SUMO_HOME'], 'tools'))
```

During each simulation episode:

- traci.start() initialised the SUMO simulation
- traci.simulationStep() advanced the simulation step-by-step
- Traffic metrics were collected through the TraCI API
- traci.close() terminated the simulation after the episode ended

SUMO was executed in headless mode during training to reduce computational overhead and accelerate experimentation. This integration enabled continuous interaction between the PPO agent and the traffic environment.

### 3.3.3 Python Environment and Required Libraries

All experiments were conducted using Python 3.10. An isolated virtual environment was created to avoid dependency conflicts and improve reproducibility:

```
python -m venv sumo_rl_env
sumo_rl_env\Scripts\activate
```

Several Python libraries were required for model training, traffic simulation control, data processing, and visualisation. The main libraries used in this work are summarised in Table 3.1 below:

Table 3.1: Required Python libraries and their roles

| Library           | Tested Version | Role in This Work                                                                                                                                       |
|-------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| stable-baselines3 | 2.3.2          | Provides the PPO implementation used for policy optimisation, including roll-out buffer management, value function training, and entropy regularisation |
| gymnasium         | 0.29.1         | Defines the standard Env interface (observation space, action space, reset/step protocol) implemented by the custom traffic environment                 |
| torch (PyTorch)   | 2.2.1          | Backend neural network library for the actor-critic policy network; supports CPU training without GPU requirement                                       |
| numpy             | 1.26.4         | Array operations for state construction, reward computation, Jain’s fairness index calculation, and metric aggregation                                  |
| pandas            | 2.2.1          | Tabular storage and manipulation of per-step metric logs produced during both training and evaluation runs                                              |
| matplotlib        | 3.8.3          | Generation of training convergence curves, fairness time series, and performance comparison charts included in Chapter 4                                |
| scipy             | 1.13.0         | Statistical analysis of evaluation results                                                                                                              |

The libraries were installed using:

```
pip install stable-baselines3 gymnasium torch numpy pandas
matplotlib scipy
```

PyTorch was used as the deep learning backend for training the PPO policy network.

### 3.3.4 Verification of the Complete Stack

After installation, the complete software stack was verified by running a short simulation test to ensure that:

- SUMO was correctly installed
- TraCI communication ran successfully
- All Python dependencies were correctly configured

- The reinforcement learning environment executed without runtime errors

A short training command was executed as a sanity check before performing the full experiments. This verification step was performed before major training experiments to ensure the consistency of the environment and prevent execution problems caused by software configuration problems. The experimental setup is summarised in Table 3.2.

Table 3.2: Experimental Environment and Configuration

| Component                       | Specification                  |
|---------------------------------|--------------------------------|
| Operating System                | Microsoft Windows 11 (64-bit)  |
| Processor                       | Intel Core i7 (6th Generation) |
| Memory                          | 16 GB RAM                      |
| Python Version                  | 3.10.x                         |
| SUMO Version                    | 1.18.0                         |
| stable-baselines3               | 2.3.2                          |
| PyTorch                         | 2.2.1 (CPU)                    |
| Simulation Duration             | 5,400 seconds per episode      |
| Control Interval ( $\Delta t$ ) | 5 seconds                      |

## 3.4 Intersection Modelling and State/Action Space Design (IM-SASD)

The simulated intersection contains multiple incoming and outgoing lanes representing the four traffic directions: north, south, east, and west, including both straight and turning vehicle movements. For traffic signal control, the intersection phases were grouped into two primary signal phases:

- East–West (EW): green phase for eastbound and westbound traffic
- North–South (NS): green phase for northbound and southbound traffic

The reinforcement learning agent dynamically selects the active traffic phase based on real time traffic conditions.

### 3.4.1 State Space

The state space contains traffic information collected from the SUMO simulator through the TraCI interface. The observed state includes:

- Queue lengths for each traffic direction
- Average vehicle waiting times for each approach

- Traffic imbalance between East–West and North–South flows
- Current traffic signal phase (EW active or NS active)

These observations allow the agent to estimate congestion levels, detect traffic build-up, and adapt signal timing decisions dynamically. The state representation is normalised to facilitate stable neural network training.

### 3.4.2 Action Space

The action space represents the centralised control decisions applied across the intersections. At each decision step (every 5 seconds), the centralised agent issues joint phase selection commands for all controlled intersections. The available discrete actions are:

- Action 0: Activate East–West (EW) green phase
- Action 1: Activate North–South (NS) green phase

This unified coordination allows network-wide maintenance of green corridors while regulating directional traffic distribution. The environment enforces minimum and maximum green time constraints to ensure realistic signal operation. The yellow phase was omitted to simplify the action space, a common simplification in RL-based traffic signal control studies [6, 12, 36]

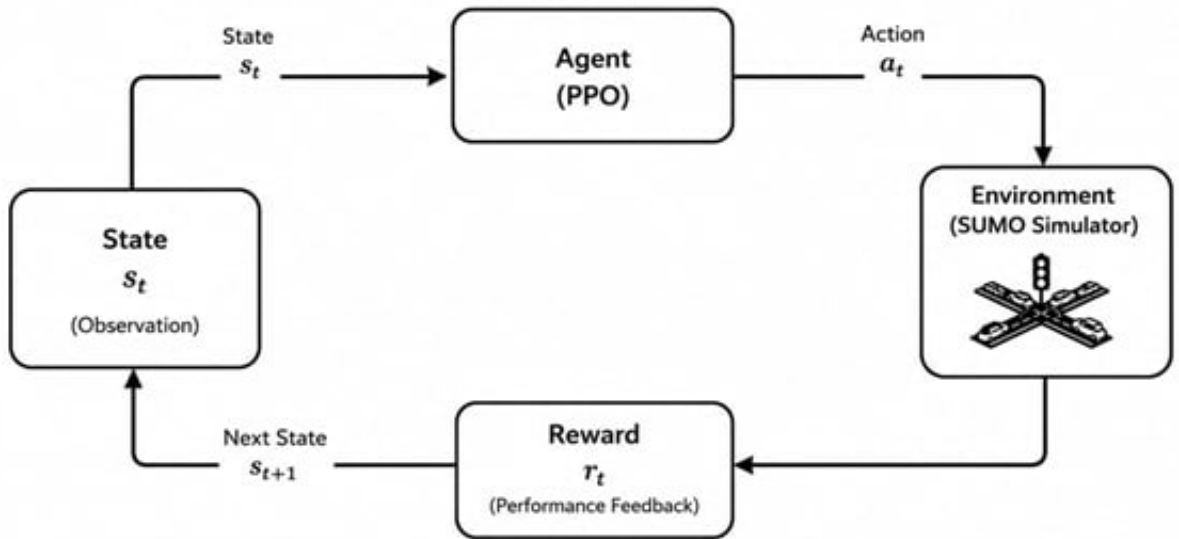


Figure 3.4: MDP Formulation for RL-Based Traffic Signal Control

As shown in Figure 3.4, the agent interacts with the environment through states, actions, and rewards.

### 3.4.3 Control Objectives

The proposed reinforcement learning controller aims to optimise both traffic efficiency and fairness. The learning process seeks to:

- Reduce vehicle waiting time across all approaches
- Reduce the length of the queue at the intersection
- Increase traffic throughput (number of vehicles served)
- Maintain fair and balanced service between traffic directions
- Minimise traffic imbalance between East–West and North–South flows

This multi-objective formulation enables the study of the trade-off between traffic efficiency and fairness in adaptive traffic signal control systems.

### 3.5 Reward Function Formulation

The reinforcement learning agent was trained using a multi-objective reward function designed to optimise both traffic efficiency and fairness within the simulated traffic network. The reward function combines several traffic performance metrics collected from the SUMO simulation environment through the TraCI interface. The reward formulation considers the following objectives:

- Minimising vehicle waiting time
- Reducing queue length
- Increasing traffic throughput
- Maintaining fairness between traffic directions
- Reducing traffic imbalance between East–West and North–South flows

The general reward formulation is expressed as:

$$R = w_1D + w_2Q + w_3T + w_4F - w_5I$$

where:

- **D** represents the delay component based on average vehicle waiting time
- **Q** represents the queue length component
- **T** represents the throughput component
- **F** represents the fairness component (Jain's Fairness Index)
- **I** represents the traffic imbalance penalty (directional waiting time asymmetry)
- **w<sub>1</sub>, w<sub>2</sub>, w<sub>3</sub>, w<sub>4</sub>, w<sub>5</sub>** are weighting coefficients used to balance the different objectives

### 3.5.1 Delay Component

The delay component rewards lower waiting times and was computed as:

$$D = \frac{1}{1 + \frac{W_{avg}}{50}} \quad (3.1)$$

where  $W_{avg}$  is the average waiting time of all vehicles currently in the network measured in seconds. This formulation produces a value between 0 and 1, with higher values indicating lower waiting times. The denominator scaling factor of 50 was chosen empirically to provide appropriate sensitivity across typical waiting time ranges.

### 3.5.2 Queue Length Component

The queue length component was defined as:

$$Q = \frac{1}{1 + \frac{Q_{total}}{150}} \quad (3.2)$$

where  $Q_{total}$  represents the total queue length across all approaches measured in number of vehicles. This component penalizes congestion by rewarding states with shorter cumulative queues. The scaling factor of 150 was selected based on the maximum expected queue length in the 2×2 network configuration.

### 3.5.3 Throughput Component

The throughput component rewards successful vehicle departures from the network and was computed as:

$$T = \frac{TP}{TP + 5} \quad (3.3)$$

where TP represents the number of arrived vehicles (vehicles that successfully passed through the intersection) during the current control interval. This component encourages the agent to prioritize signal phases that serve more vehicles, contributing to overall network efficiency.

### 3.5.4 Fairness Component (Jain's Fairness Index)

Fairness between traffic directions was evaluated using Jain's Fairness Index:

$$F = \frac{(W_{EW} + W_{NS})^2}{2(W_{EW}^2 + W_{NS}^2)} \quad (3.4)$$

where:

- $W_{EW}$  is the network-wide aggregated average waiting time for all East–West movements across the grid
- $W_{NS}$  is the network-wide aggregated average waiting time for all North–South movements across the grid

A fairness value close to 1 indicates balanced traffic service between directions, meaning both directions experience similar waiting times. Lower values indicate unequal signal allocation where one direction receives preferential treatment while the opposite direction experiences excessive waiting times.

### 3.5.5 Imbalance Penalty

To discourage excessive preference toward one traffic direction, an imbalance penalty was introduced:

$$I = \frac{|W_{EW} - W_{NS}|}{W_{EW} + W_{NS}} \quad (3.5)$$

This penalty term increases when waiting times diverge between the two traffic directions, encouraging the agent to maintain balanced service even under asymmetric traffic demand conditions.

### 3.5.6 Weight Configurations

Different reward weight combinations were tested during training to study the trade-off between traffic efficiency and fairness. Some models emphasized traffic efficiency by assigning higher weights to delay reduction ( $w_1, w_3$ ) and throughput improvement, while other models prioritized fairness metrics ( $w_4$ ) and imbalance reduction.

After multiple training experiments and reward adjustments, a balanced reward configuration was obtained. The final reward design enabled the RL agent to improve traffic efficiency while maintaining fair service distribution between traffic directions.

Table 3.3: Reward weight configurations for the evaluated models

| Configuration          | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ |
|------------------------|-------|-------|-------|-------|-------|
| Balanced               | 0.30  | 0.20  | 0.25  | 0.25  | 0.15  |
| Fairness-Oriented      | 0.20  | 0.20  | 0.15  | 0.45  | 0.15  |
| Efficiency-Oriented    | 0.35  | 0.15  | 0.35  | 0.15  | 0.15  |
| Vanilla PPO (Baseline) | 1.00  | 0.30  | 0.20  | 0.00  | 0.00  |

Table 3.3 outlines the empirical reward weight configurations designed to evaluate the trade-offs between traffic efficiency and service equity across the five tested controller variations.

To ensure mathematical consistency and fair behavioural comparison during the reinforcement learning gradient ascent optimisation, all raw weight vectors ( $\mathbf{w}_{raw} = [w_1, w_2, w_3, w_4, w_5]$ ) are dynamically normalised within the environment implementation so that the efficiency-related components satisfy the unity constraint  $\bar{w}_1 + \bar{w}_2 + \bar{w}_3 + \bar{w}_4 = 1.0$ . The imbalance penalty  $w_5$  is applied separately without normalisation.

For instance, while the baseline Vanilla PPO is configured with raw values of  $w_1 = 1.00$ ,  $w_2 = 0.30$ , and  $w_3 = 0.20$  (accumulating to a raw total of 1.50), the intrinsic normalisation procedure scales these components relative to the total mass of the vector. This transformation yields effective operational weights of approximately  $\bar{w}_1 \approx 0.667$  for delay reduction,  $\bar{w}_2 \approx 0.200$  for queue minimisation, and  $\bar{w}_3 \approx 0.133$  for global throughput maximisation, while strictly maintaining zero constraints on fairness and imbalance ( $\bar{w}_4 = 0.00$ ,  $\bar{w}_5 = 0.00$ ).

This proportional scaling guarantees that the magnitude of the absolute reward signal remains bounded and standardised across all experimental variants. Consequently, the observed behavioural shifts among the controllers (such as policy crystallisation or emergent equity) are purely driven by the directional distribution and relative ratios of the weight objectives rather than artificial variations in reward scalar gradients.

## 3.6 Reinforcement Learning Algorithm Choice

### 3.6.1 Overview of Proximal Policy Optimisation (PPO)

Proximal Policy Optimisation (PPO) was selected as the reinforcement learning algorithm used in this work. PPO is a policy gradient method introduced by OpenAI in 2017 to provide stable and efficient policy updates while maintaining relatively low implementation complexity compared to earlier methods like Trust Region Policy Optimisation (TRPO) [20]. Unlike value-based reinforcement learning algorithms (such as DQN) that learn action values and derive policies indirectly, PPO directly optimises the policy followed by the agent [15, 20].

The algorithm belongs to the family of actor–critic methods [15, 20], where two neural networks operate simultaneously during training:

- **The actor network** selects actions according to the observed traffic state, learning which signal phase to activate under different congestion conditions [20].
- **The critic network** estimates the expected future reward (value function) of the current policy, providing feedback to improve the actor’s decisions [20].

This actor–critic structure enables the agent to improve traffic signal decisions while simultaneously learning the long-term quality of those decisions [15, 20]. PPO was particularly suitable for adaptive traffic signal control because traffic conditions evolve continuously over time and require dynamic policy adaptation rather than fixed rule-based control strategies [12, 20].

### 3.6.2 The PPO Objective Function

The PPO objective function includes a clipped policy update mechanism that limits excessively large policy changes during training. The core objective is:

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[ \min \left( r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right] \quad (3.6)$$

where:

- $r_t(\theta)$  is the probability ratio between the new and old policies
- $\hat{A}_t$  is the estimated advantage function
- $\epsilon$  is the clipping hyperparameter (set to 0.2 in this work)

This clipping mechanism improves learning stability and prevents destructive updates that could degrade traffic control performance.

### 3.6.3 Advantages of PPO for Traffic Signal Control

PPO provides several advantages that make it appropriate for adaptive traffic signal control problems.

**First**, PPO offers stable training behaviour through clipped policy optimisation [20]. Traffic environments are highly dynamic and stochastic, influenced by random vehicle arrivals, driver behaviour variations, and complex vehicle interactions [12, 25]. Unstable policy updates can easily produce oscillatory or degenerate traffic signal behaviour. PPO reduces this risk by constraining policy updates within a safe optimisation range[20].

**Second**, PPO naturally balances exploration and exploitation during learning[20]. Exploration enables the agent to test alternative traffic signal decisions, discovering potentially better control strategies. Exploitation allows the model to use previously learned successful policies. This balance is particularly important in traffic control because traffic distributions may vary significantly across different simulation episodes and times of day[12].

**Third**, PPO supports continuous interaction with large and complex state spaces.

The traffic environment used in this thesis includes multiple traffic metrics such as queue lengths, waiting times, throughput, fairness indicators, and traffic imbalance measures. PPO handles these multidimensional observations effectively through neural-network-based policy learning without requiring manual feature engineering[20].

PPO was preferred over Deep Q-Network (DQN) approaches because DQN often suffers from unstable learning behaviour in highly dynamic environments with delayed rewards and complex state interactions[15, 20]. The combination of policy-gradient optimisation, actor–critic learning, and clipped policy updates made PPO well suited for the multi-objective traffic optimisation problem addressed in this thesis.

## 3.7 RL Model Implementation

### 3.7.1 Framework Implementation

The reinforcement learning framework was implemented using the Stable-Baselines3 library integrated with PyTorch as the deep learning backend. Stable-Baselines3 was used because it provides reliable and optimised implementations of modern reinforcement learning algorithms, including PPO, together with support for logging, checkpoint management, and policy evaluation.

The traffic signal control environment was implemented using the Gymnasium framework, which provides a standardised interface for reinforcement learning environments, including:

- State observation handling with consistent data types and shapes
- Action execution with validation and error handling
- Reward computation using the multi-objective reward function
- Episode management including reset and termination conditions

The custom Gymnasium environment was connected to the SUMO traffic simulator through the TraCI interface, allowing the PPO agent to interact with the traffic environment in real time.

### 3.7.2 Neural Network Architecture

The PPO agent used an actor–critic neural network architecture consisting of fully connected feed forward layers. The policy network architecture contained three hidden layers:

256 → 256 → 128

Rectified Linear Unit (ReLU) activation functions were used between hidden layers to introduce non-linearity and improve feature learning capability. The output layer of the actor network used a softmax activation to produce a probability distribution over the two available traffic signal actions. The critic network output was a single scalar value representing the estimated state value.

This architecture was selected because it provided sufficient representational capacity to learn complex traffic control policies while maintaining stable training performance and avoiding overfitting to specific traffic patterns.

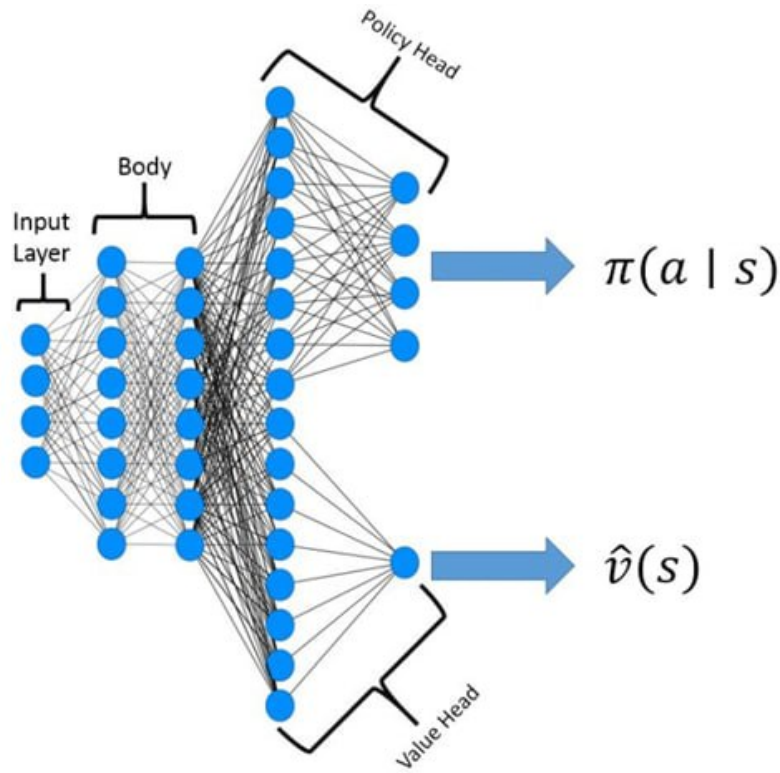


Figure 3.5: PPO Actor-Critic Neural Network Architecture [34]

In this Figure 3.5 the PPO model uses a shared neural network backbone followed by separate actor and critic heads. The actor network outputs the traffic signal control policy, while the critic network estimates the state value used for policy optimisation.

### 3.7.3 Logging and Checkpoint System

A logging and checkpoint mechanism was implemented during training to monitor reinforcement learning performance continuously. During training, multiple traffic performance metrics were recorded at regular intervals, including:

- Episode rewards (cumulative and average)
- Queue lengths (total and per approach)
- Waiting times (average and maximum)
- Throughput (vehicles served per episode)
- Jain's fairness index (per control interval and episode average)
- Traffic imbalance indicators
- Policy loss and value loss values

Performance metrics were periodically stored in CSV format for later analysis and visualisation using pandas and matplotlib.

Checkpoint saving was also implemented during training. Whenever the PPO agent achieved improved reward performance (evaluated using a moving average window), the current model parameters were automatically saved as a candidate best model. This checkpoint system reduced the risk of losing high-performing policies during long training sessions and enabled later comparison between different reward configurations and training seeds.

### 3.7.4 Evaluation Strategy

Model evaluation was performed using deterministic policy execution to ensure reproducible testing behaviour. During evaluation, the PPO agent selected actions deterministically (taking the action with highest probability) without stochastic exploration in order to measure the learned traffic control policy consistently across multiple evaluation runs.

The trained models were evaluated under multiple traffic scenarios, including:

- Imbalanced traffic flows (higher demand on one axis)
- Fairness-oriented traffic conditions
- Efficiency-oriented traffic conditions
- Unseen traffic routes (generalisation testing)

Evaluation metrics collected during this phase included:

- Average waiting time (seconds per vehicle)
- Queue length (number of vehicles)
- Throughput (vehicles served per hour or per episode)
- Travel time (total time from arrival to departure)
- Jain's fairness index (service balance between directions)

The evaluation methodology enabled detailed comparison between different reward configurations and baseline traffic control methods.

## 3.8 Training Procedure and Experimental Configuration (TP-EC)

### 3.8.1 Training Procedure

The PPO agent was trained through repeated interaction with the SUMO traffic simulation environment. Each training episode followed this sequence:

- 1. Environment initialisation:** The SUMO simulator loads the network file, route configuration, and traffic signal parameters. The TraCI connection is established.
- 2. State observation:** Traffic observations (queue lengths, waiting times, current phase) are collected from SUMO through TraCI.
- 3. Action selection:** The PPO agent's actor network processes the observations and selects a traffic signal action (EW green or NS green) with stochastic exploration.
- 4. Action execution:** The selected action is transmitted to SUMO through TraCI and applied to the traffic signal controller for the 5-second control interval.
- 5. Simulation advance:** SUMO advances the simulation by one step (5 seconds), updating vehicle positions, queues, and waiting times.
- 6. Reward computation:** A reward value is computed using the proposed multi-objective reward function based on the observed traffic outcomes.
- 7. Experience storage:** The transition (state, action, reward, next state) is stored in the PPO agent's roll-out buffer.
- 8. Policy update:** At regular intervals, the agent performs PPO policy optimisation using the collected experiences.

This interaction loop was repeated continuously throughout the simulation episode until 5,400 simulation seconds (90 minutes of simulated traffic time) were completed, resulting in 1,080 control decisions per episode (given the 5-second control interval). Each training experiment utilised approximately 300,000 total interaction steps, equivalent to about 278 complete episodes ( $300,000 \div 1,080 \approx 277.8$  policy updates) to ensure comprehensive convergence.

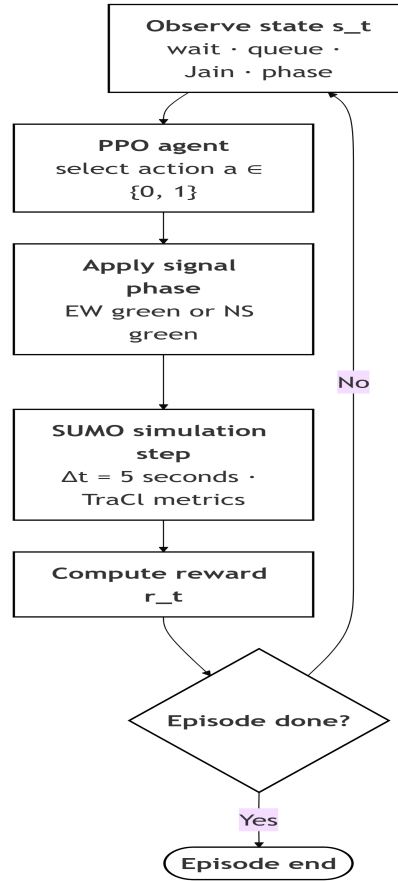


Figure 3.6: Decision-making loop of the PPO-based traffic signal controller.

This Figure 3.6 shown at each 5-second control step, the agent observes the traffic state, selects a signal phase, advances the SUMO simulation by the 5-second control interval (delta-time = 5), computes the multi-objective reward, and checks for episode termination. The loop repeats for up to 1,080 steps per episode (90 minutes).

### 3.8.2 Hyperparameter Settings

Table 3.4 summarises the main PPO hyperparameters used during training to improve training stability and learning performance.

Table 3.4: PPO Hyperparameter Configuration

| Hyperparameter                | Value              | Description                                                   |
|-------------------------------|--------------------|---------------------------------------------------------------|
| Learning Rate                 | $3 \times 10^{-4}$ | Step size for policy and value network updates                |
| Discount Factor ( $\gamma$ )  | 0.99               | Weight placed on future rewards relative to immediate rewards |
| Batch Size                    | 256                | Number of experiences sampled per gradient update             |
| PPO Epochs                    | 10                 | Number of optimisation passes over the same batch             |
| Entropy Coefficient           | 0.02               | Encourages exploration by adding entropy bonus to the loss    |
| Clipping Range ( $\epsilon$ ) | 0.2                | Constrains policy updates to improve stability                |
| GAE Lambda ( $\lambda$ )      | 0.95               | Bias-variance trade-off for advantage estimation              |
| Maximum Gradient Norm         | 0.5                | Clips gradients to prevent exploding gradient problems        |

These hyperparameters were selected based on reinforcement learning best practices reported in the literature and preliminary experimental testing on the traffic environment.

### 3.8.3 Multi-Seed Training

To improve robustness and reduce stochastic variance, multiple random seeds were used during training. Different seeds affected:

- PPO neural network parameter initialisation
- Environment initialisation (SUMO random number generator)
- Training trajectory order and exploration behaviour
- Vehicle arrival patterns (within the same statistical distribution)

A total of five random seeds were used during experimentation (seeds 42, 123, 456, 789, and 1024). Training with multiple seeds improved reproducibility and reduced the probability of obtaining biased results caused by favourable random initialisation conditions.

#### Note

Vanilla PPO was trained with a single seed (789) as a deterministic baseline, and Fixed-Time (non-adaptive) requires no training.

### 3.8.4 Pareto-Based Experiments

Multiple reward-weight configurations were investigated to study the trade-off between traffic efficiency and fairness. The three configurations (fairness-oriented, balanced, and efficiency-oriented) are detailed in Table 3.3 (see Section 3.5.6). Different experiments emphasised different optimisation objectives:

**Fairness-oriented reward configurations:** Higher weights assigned to  $w_4$  (fairness component) and  $w_5$  (imbalance penalty)

**Efficiency-oriented reward configurations:** Higher weights assigned to  $w_1$  (delay reduction),  $w_2$  (queue reduction), and  $w_3$  (throughput)

**Balanced multi-objective configurations:** Moderate weights across all components

These experiments enabled analysis of the relationship between congestion reduction, throughput optimisation, and fair traffic service distribution. The experiments demonstrated that maximising efficiency alone often reduced fairness (causing one direction to experience excessive waiting times), while excessive fairness prioritisation could increase congestion and overall waiting time.

The complete training campaign, comprising 15 configurations (5 random seeds  $\times$  3 reward configurations), required approximately 72 hours (3 full days) of continuous real-time computation on the hardware platform specified in Table 3.2. Each configuration was trained for 300,000 time steps, resulting in a total of 4.5 million time steps across all experiments.

### 3.8.5 Best Model Selection

A total of fifteen PPO training runs were conducted using different random seeds and reward configurations (5 seeds  $\times$  3 reward configurations). The best-performing models were selected according to several evaluation metrics, including:

- Cumulative reward performance during validation episodes
- Waiting time reduction (average and maximum)
- Queue length reduction (total and per approach)
- Throughput improvement (vehicles served per episode)
- Fairness performance (Jain’s index average and minimum)
- Policy stability (variance across different evaluation episodes)

After training all five seeds for each reward configuration, the best-performing seed per variant was selected based on its primary objective. The results reported in Chapter 4 represent this optimal seed; consequently, performance variability across seeds is not fully captured (see Section 4.6 for a discussion of this limitation).

## 3.9 Baseline Implementation

### 3.9.1 Fixed-Time Controller

A fixed-time traffic signal controller was implemented as a traditional baseline method representing conventional traffic signal operation [14, 16]. The controller operates using predefined signal durations without considering real-time traffic conditions. Traffic phases alternate according to constant timing intervals throughout the simulation. The timing parameters were: 125 seconds of East-West (EW) green, followed by 125 seconds of North-South (NS) green, repeating throughout the simulation. This corresponds to 25 control steps at a 5-second control interval (since  $\Delta t = 5$  seconds). No yellow phase was used, as the controller alternates directly between the two main phases. This baseline enabled comparison between adaptive reinforcement learning control and conventional static traffic signal operation, establishing a lower performance bound for the evaluation.

### 3.9.2 Vanilla PPO

A Vanilla PPO model was also implemented for comparison purposes. Unlike the proposed multi-objective fairness-aware PPO model, the Vanilla PPO focused primarily on traffic efficiency metrics such as waiting time reduction and queue length minimisation. The baseline model used the same neural network architecture and hyperparameters as the proposed model, but its reward function omitted the fairness component ( $w_4=0$ ) and imbalance penalty ( $w_5=0$ ). As a result, the baseline model learned to prioritise the direction with higher traffic demand without considering fairness between competing directions. This comparison enabled analysis of the impact of fairness-aware reward design on traffic signal performance, specifically whether incorporating fairness objectives reduced efficiency or improved overall system performance through balanced service.

### 3.9.3 Comparison Methodology

Comparative evaluation was conducted between:

- Fixed-time traffic control
- Baseline PPO control (efficiency-only reward)
- Proposed fairness-aware PPO models (three configurations)

All controllers were evaluated under identical traffic scenarios and simulation conditions, including the same vehicle arrival distributions, traffic demands, and random seeds. Each experiment was repeated five times with different random seeds to obtain statistically significant results.

Performance comparison was based on:

- Average waiting time (seconds per vehicle)
- Queue length (maximum and average vehicles)
- Throughput (vehicles served per hour)
- Travel time (seconds per vehicle from arrival to departure)
- Jain’s fairness index (service balance between EW and NS)

## 3.10 Experimental Scenarios

### 3.10.1 Imbalanced Traffic Scenario

Imbalanced traffic scenarios introduced unequal traffic demand between competing traffic directions [21, 22]. Specifically, the East–West direction received 600 vehicles per hour while the North–South direction received 200 vehicles per hour, creating a fixed 3 : 1 structural demand ratio during the training phase. These experiments simulated realistic urban traffic situations where one corridor experiences significantly higher traffic volume than the other, such as a main arterial road intersecting with a minor side street, or directional traffic peaks during morning and evening rush hours. The structurally imbalanced scenarios were particularly important for evaluating fairness behaviour and adaptive traffic prioritisation [10]. An effective controller should dynamically provide more green time to the high-demand direction while still serving the low-demand direction equitably without inducing excessive delay accumulation or queue saturation.

### 3.10.2 Fairness-Oriented Experiments

Fairness-oriented experiments used reward configurations that assigned higher empirical weights to equity-related objectives (where the fairness weight  $w_4$  and the imbalance penalty weight  $w_5$  were increased, while the delay  $w_1$ , queue  $w_2$ , and throughput  $w_3$  weights were decreased proportionally) [10, 21]. These experiments aimed to investigate whether a deep reinforcement learning agent could maintain balanced and non-discriminatory traffic service between competing phases even under severely imbalanced demand conditions. The fairness-oriented configuration (specifically the PPO Fairness and PPO Balanced profiles detailed in Table 3.3) served as an operational benchmark for equity performance, evaluating the maximum achievable service balance across intersections regardless of potential localised throughput costs [10, 21].

### 3.10.3 Efficiency-Oriented Experiments

Efficiency-oriented experiments prioritised global traffic performance metrics, specifically focusing on delay reduction, queue minimisation, and throughput maximisation [6]. These experiments established the operational baselines and performance upper bounds for

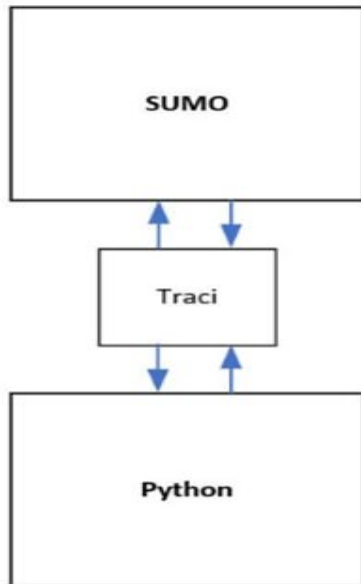
pure traffic efficiency without the integration of fairness constraints ( $w_4 = 0.00$  and  $w_5 = 0.00$ ). This phase evaluated two distinct configurations: first, the Vanilla PPO (Baseline), which utilises a multi-objective efficiency reward structure optimised with weights  $w_1$  (Delay) = 1.00,  $w_2$  (Queue) = 0.30, and  $w_3$  (Throughput) = 0.20; and second, the specialised PPO Efficiency configuration optimised heavily for maximum vehicle clearance. These experiments mapped the maximum capabilities of the PPO agent under heavy, structurally imbalanced traffic conditions, providing a critical benchmark against which the proposed fairness-aware models are compared.

### 3.10.4 Unseen Route Evaluation

To evaluate zero-shot generalisation capability, the trained PPO policies were tested on an unseen traffic route layout (unseen-route.xml) that was completely omitted during the training phase [20]. In this evaluation, the demand profile was modified to introduce a temporally-varying symmetric distribution shift. While individual phases within the episode exhibit severe local imbalances (where North-South demand dominates the first 1,800 seconds and East-West demand dominates the subsequent 1,800 seconds), the cumulative global volume remains mathematically balanced and symmetric across the entire 5,400-second duration. This setup evaluated whether the learned traffic signal policies could adapt to sudden directional inversions and dynamic demand transitions without additional online retraining [20]. Successful performance on this unseen layout indicates that the agent has captured generalised principles of traffic state-action dynamics (such as real-time queue clearance) rather than memorising the fixed spatial asymmetry of the training route. These experiments provide vital insights into the robustness and transferability of the proposed multi-objective reinforcement learning framework under unfamiliar deployment conditions, which is essential for real-world traffic networks characterised by non-periodic variations across days and seasons [10, 12].

## 3.11 Integration Flow Between RL Agent and Simulator (IF-RLAS)

### 3.11.1 Integration Architecture



The proposed reinforcement learning framework was implemented through the integration of the SUMO traffic simulator with Python using the Traci (Traffic Control Interface) communication protocol.

Figure 3.7: Integration Architecture Between SUMO, Traci, and Python [35]

The Figure 3.7 illustrates the communication architecture used to connect the SUMO traffic simulator with the Python-based reinforcement learning framework through the Traci interface.

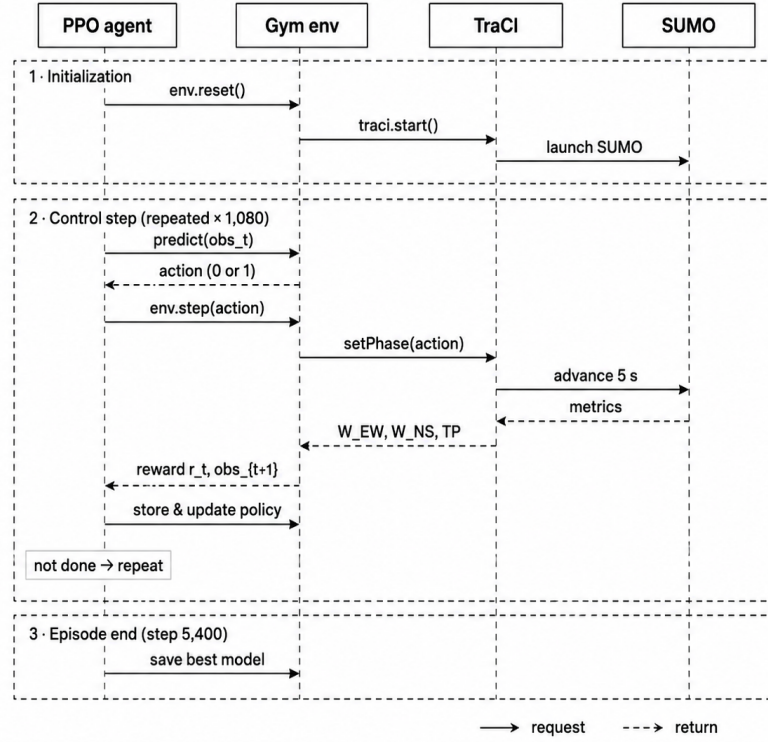


Figure 3.8: Interaction sequence between the training script, PPO agent, Gymnasium environment, TraCI, and SUMO simulator

Figure 3.8 illustrates the interaction loop between the PPO agent and the SUMO environment. Solid arrows represent request (function call) messages, while dashed arrows represent return messages (observations, rewards, and termination signals). The control loop repeats 1,080 times per episode with a 5-second control interval.

### 3.11.2 Simulation Initialisation

At the beginning of each simulation episode, the SUMO simulator initialises the urban traffic network, traffic signal configurations, vehicle routes, and simulation parameters. The simulator then starts execution through the TraCI interface using:

```
traci.start(["sumo", "-c", "sumo_config.sumocfg"])
```

This initialisation creates the communication channel between Python and SUMO and prepares the environment for reinforcement learning interaction.

### 3.11.3 State Observation Collection

During each control interval, traffic state information is collected from SUMO through the TraCI API using specific function calls. Table 3.5 summarises the traffic observation metrics

and their corresponding TraCI functions used for state collection.

Table 3.5: Traffic observation metrics and corresponding TraCI functions

| Observation                  | TraCI Function / Method                          |
|------------------------------|--------------------------------------------------|
| Vehicle waiting times        | <code>traci.vehicle.getWaitingTime()</code>      |
| Queue lengths                | Vehicles with speed < 0.1 m/s near the stop line |
| Current traffic signal phase | <code>traci.trafficlight.getPhase()</code>       |
| Throughput                   | <code>traci.simulation.getArrivedIDList()</code> |

These traffic measurements are then converted into the state representation used by the PPO reinforcement learning agent. The resulting state representation enables the agent to estimate current congestion conditions and evaluate traffic distribution across different directions.

### 3.11.4 RL Decision-Making Process

After receiving the current traffic state, the PPO agent processes the observations using the actor–critic neural network architecture. The state vector is passed through the three hidden layers ( $256 \rightarrow 256 \rightarrow 128$ ) with ReLU activations, producing logits for the two possible actions.

Based on the current policy (with stochastic exploration during training or deterministic action selection during evaluation), the agent selects the next traffic signal action. The available actions correspond to activating one of the traffic signal phases:

- **Action 0:** East–West (EW) green phase
- **Action 1:** North–South (NS) green phase

### 3.11.5 Traffic Signal Execution

Once the action is selected, the new traffic signal phase is transmitted back to the SUMO simulator through the TraCI interface using:

```
traci.trafficlight.setPhase(tls_id, action)
```

The simulator updates the active traffic signal phase and executes the selected control action for the predefined control interval (5 seconds). During this period:

- Vehicle movements are updated according to traffic flow models
- Queues evolve dynamically as vehicles arrive and depart
- Waiting times change as vehicles remain stopped or begin moving

- New vehicles enter and leave the network according to route files

The simulation progresses step-by-step using:

```
traci.simulationStep()
```

This process enables continuous interaction between the traffic environment and the RL controller.

### 3.11.6 Reward Computation and Learning Feedback

After executing the selected action and advancing the simulation by one control interval, the environment computes a reward value using the proposed multi-objective reward function described in Section 3.5. The reward combines several traffic performance objectives:

- Waiting time reduction (D component)
- Queue length reduction (Q component)
- Throughput improvement (T component)
- Fairness preservation (F component using Jain's index)
- Traffic imbalance reduction (I penalty)

The computed reward and updated traffic state are returned to the PPO agent for policy learning and optimisation. This feedback mechanism allows the agent to gradually improve its traffic signal control strategy through repeated interaction with the environment.

### 3.11.7 Continuous Closed-Loop Interaction

The interaction cycle between the RL agent and the SUMO simulator is repeated continuously throughout the entire simulation episode. The complete interaction loop can be summarised as:

1. SUMO initialises the traffic environment
2. Traffic observations are collected through TraCI
3. The RL agent receives the current state vector
4. PPO selects the next traffic signal action
5. The selected action is applied in SUMO
6. Traffic conditions evolve dynamically over 5 seconds
7. A reward value is computed based on observed outcomes
8. The new state and reward are returned to the RL agent
9. The experience is stored and used for policy updates

This closed-loop framework enables the reinforcement learning agent to gradually learn adaptive traffic signal control policies capable of responding to dynamic traffic conditions without requiring explicit traffic models or manual parameter tuning.

### 3.11.8 Real-Time Communication Benefits

The integration between SUMO, Python, and the PPO agent enabled:

- Real-time traffic signal control with 5-second adaptation intervals
- Dynamic adaptation to changing traffic conditions and demand patterns
- Continuous reward-based policy learning through millions of interaction steps
- Evaluation under multiple traffic scenarios ( imbalanced, unseen routes)
- Fairness and efficiency trade-off analysis using different reward configurations
- Generalisation testing using unseen traffic routes without additional training

The modular integration architecture also improved the flexibility and reproducibility of the experimental framework developed in this thesis, allowing easy modification of reward functions, network architectures, and simulation parameters across different experimental conditions.

## 3.12 Chapter Summary

This chapter presented the complete system architecture, methodology, and experimental framework for the reinforcement learning-based adaptive traffic signal control system. The key components described include:

- **Simulation environment:** SUMO-based microscopic traffic simulation with TraCI real-time communication, including a custom 2×2 urban road network, traffic demand generation through route files, and unseen route evaluation for generalisation testing.
- **Experimental configuration:** Complete software stack including SUMO 1.18.0, Python 3.10, stable-baselines3 2.3.2, and PyTorch 2.2.1, with hardware specifications documented for reproducibility.
- **State and action space design:** A state space comprising queue lengths, waiting times, traffic imbalance, and current phase, with a two-action discrete action space (EW green or NS green).
- **Multi-objective reward function:** A weighted combination of delay reduction, queue minimisation, throughput maximisation, fairness preservation using Jain’s Fairness Index, and imbalance penalty, with flexible weight configurations for efficiency-fairness trade-off analysis.
- **Reinforcement learning algorithm:** Proximal Policy Optimisation (PPO) selected for its stable training behaviour, natural exploration-exploitation balance, and suitability for complex state spaces.
- **Training procedure:** Multi-seed training with five random seeds, hyperparameter config-

uration (learning rate  $3 \times 10^{-4}$ , discount factor 0.99, batch size 256), and best model selection from fifteen trained configurations.

- **Baseline methods:** Fixed-time controller and efficiency-only PPO controller for comparative evaluation against the proposed fairness-aware models.
- **Experimental scenarios:** imbalanced traffic (3:1 ratio), fairness-oriented experiments, efficiency-oriented experiments , and unseen route generalisation testing .

The experimental framework established in this chapter provides the foundation for the results presented in Chapter 4.

# Chapter 4

## Results and Analysis

### 4.1 Introduction

This chapter presents the experimental evaluation and performance analysis of the proposed reinforcement learning-based traffic signal control framework using Proximal Policy Optimisation (PPO). The study compares five traffic signal controllers: Fixed-Time Baseline, Vanilla PPO, PPO Fairness, PPO Balanced, and PPO Efficiency. Each controller is evaluated under two different traffic demand scenarios, namely the Training Route and the Unseen Route, in order to assess both learning performance and generalisation capability.

The evaluation focuses on several key traffic performance metrics, including waiting time, queue length, throughput, travel time, and fairness. Through detailed experimental analysis and comparative visualisation, the chapter investigates the strengths, weaknesses, and behavioural trade-offs of each controller under varying traffic conditions.

In addition, this chapter examines the impact of reward-function design on controller behaviour, particularly the trade-off between traffic efficiency and directional fairness. generalisation experiments are also conducted to evaluate robustness, adaptability, and deployment reliability of the proposed PPO-based approaches.

### 4.2 Experimental Evaluation Setup

#### 4.2.1 Simulation Environment

All experiments were conducted using the SUMO (Simulation of Urban MObility) traffic simulator, an open-source microscopic traffic simulation platform capable of modelling individual vehicle dynamics, lane-level interactions, and traffic signal operations with high fidelity. The simulated road network consists of a  $2 \times 2$  signalised urban grid comprising four traffic-light-controlled intersections (IDs: 1, 2, 5, and 6), interconnected through bidirectional horizontal and vertical arterial roads, forming a compact urban grid topology. Each

road segment contains two lanes per direction, with a maximum vehicle speed of 13.89 m/s (approximately 50 km/h).

Two route files were designed to evaluate both learning performance and generalisation capability.

The file `2x2.rou.xml` defines the Training Route, characterised by a static, structurally imbalanced traffic distribution in which East–West flows are significantly heavier than North–South flows (approximately a 3:1 ratio).

In contrast, the file `unseen-route.xml` defines the Unseen Route, which introduces a temporally varying symmetric distribution shift. Vehicles are generated during the first 3,600 seconds of each episode. Within this generation period, North–South demand dominates the first 1,800 seconds, while East–West demand dominates the remaining 1,800 seconds. Consequently, the total generated demand remains globally balanced and symmetric, despite the inversion of directional dominance midway through the generation period. This temporal symmetry serves as an effective dual-purpose benchmark: it tests reinforcement learning policies against sudden distribution inversions while providing an environment in which traditional static control splits may perform competitively due to the global balance of traffic demand.

Each simulation episode continues for a total duration of 5,400 seconds (90 minutes) to allow network clearance and congestion dissipation after vehicle generation ceases. The reinforcement learning agent interacts with the environment every 5 seconds ( $\text{delta-time} = 5$ ), resulting in up to 1,080 control decisions per episode. To preserve realistic congestion dynamics, vehicle teleportation was disabled using the SUMO parameter `–time-to-teleport -1`, ensuring that vehicles remain physically queued regardless of delay accumulation. Additionally, the number of vehicles per episode was capped at 5,000 to prevent memory overflow during extended simulations.

Multiple random seeds were used during the training phase to improve policy robustness and reduce sensitivity to stochastic initialisation. Following training, the best-performing policies were selected based on multiple evaluation metrics and subsequently used for the final performance evaluation and comparative analysis.

### 4.2.2 Evaluated Controllers

Five controllers spanning the spectrum from non-adaptive baselines to reward-shaped RL agents are compared:

1. **Fixed-Time Baseline:** A conventional pre-timed controller applying a symmetric green interval of 25 control steps (corresponding to 125 seconds based on a 5-second delta time interval) alternately to the East-West (EW) and North-South (NS) traffic phases. It performs no learning, stores no state, and applies identical timing regardless of real-time traffic conditions. It serves as the primary reference point for evaluating the added

value of adaptive RL control.

2. **Vanilla PPO:** A standard PPO agent trained with a reward function emphasizing delay reduction (weight: delay = 1.0) with secondary contributions from queue length (0.3) and throughput (0.2), and zero weight on fairness (0.0). After normalization, the effective contribution proportions are delay  $\approx 0.667$ , queue  $\approx 0.200$ , throughput  $\approx 0.133$ . This controller represents the pure efficiency-oriented RL baseline.
3. **PPO Fairness:** A fairness-optimised PPO variant assigning the highest reward weight to Jain's Fairness Index (fairness = 0.45). The remaining weights (delay, queue, throughput) are proportionally reduced to accommodate the fairness emphasis, plus an imbalance penalty of 0.15 discouraging systematic directional discrimination. This controller is designed to produce equitable service across all traffic directions.
4. **PPO Balanced:** A multi-objective PPO variant combining delay (0.30), queue (0.20), throughput (0.25), and fairness (0.25) at comparable weight levels, with an imbalance penalty of 0.15. The design intent is to achieve a productive compromise between efficiency and fairness without sacrificing either to an extreme.
5. **PPO Efficiency:** A throughput-maximizing PPO variant prioritising vehicle flow (throughput = 0.35) and delay reduction above all other objectives. Fairness receives a minimal weight, positioning this controller as the most aggressive efficiency optimiser in the comparison set.

### 4.2.3 Evaluation Metrics

Six metrics characterise controller performance across efficiency, equity, and congestion dimensions:

- **Average Waiting Time (s)** (lower is better): Mean time vehicles spend stationary at a red phase across all network edges within one episode. Directly reflects driver delay experience.
- **Average Queue Length (vehicles)** (lower is better): Mean number of vehicles queued across all road segments per control time-step. Captures sustained congestion load on the network.
- **Total Throughput (vehicles/episode)** (higher is better): Cumulative count of vehicles that successfully departed the simulation network within one episode. Primary indicator of network utilisation efficiency.
- **Jain's Fairness Index** (higher is better): Computed as  $J = (\sum x_i)^2 / (n \cdot \sum x_i^2)$  where  $x_i$  denotes the directional waiting time for each traffic axis (EW, NS). Ranges from

0 (maximally unequal) to 1 (perfectly balanced). A value of  $\geq 0.8$  is adopted as the minimum acceptable fairness threshold in this study.

- **Average Travel Time (s)** (lower is better): Mean elapsed time from vehicle insertion to departure, encompassing both movement time and cumulative waiting time at signals.
- **Episode Reward** (training diagnostics only): Mean per-step reward received by the RL agent, normalized and clipped to  $[-1, +1]$  by design. Used to monitor training convergence; not directly interpretable as a performance metric on objective-level traffic outcomes.

#### 4.2.4 Evaluation Scenarios

Two evaluation scenarios assess controller performance:

- **Training Route**: The demand profile (`2x2.rou.xml`) used for training all PPO controllers.
- **Unseen Route**: A structurally distinct demand profile (`unseen_route.xml`) not observed during training. Evaluated in zero-shot transfer (no retraining or policy adjustment).

### 4.3 Comparative Performance on Training Route

#### 4.3.1 Training Route Performance Summary

Table 4.1 consolidates all six performance metrics for the five controllers on the Training Route, reward values are not comparable across controllers due to different reward function weights (see Section 4.3.7).

Table 4.1: Training Route Performance Summary

| Controller     | Wait (s)<br>↓ | Queue (veh) ↓ | Throughput ↑ | Jain’s Index ↑ | Travel (s) ↓ | Reward ↑ |
|----------------|---------------|---------------|--------------|----------------|--------------|----------|
| Fixed-Time     | 29.64         | 182.9         | 462          | —              | 339.1        | -0.115   |
| Vanilla PPO    | 16.94         | 171.9         | 634          | —              | 336.5        | +0.220   |
| PPO Fairness   | 17.82         | 177.6         | 459          | 0.9473         | 328.7        | +0.244   |
| PPO Balanced   | 16.62         | 168.2         | 545          | 0.9385         | 337.9        | +0.025   |
| PPO Efficiency | 15.63         | 170.8         | 655          | 0.9304         | 278.4        | -0.118   |

All reported values are averaged over five independent simulation runs. For PPO variants (Fairness, Balanced, and Efficiency), each policy was trained using five different random seeds. The best-performing seed for each variant was selected based on its primary optimization objective and subsequently evaluated over five simulation runs using different traffic seeds. In contrast, the Fixed-Time baseline and Vanilla PPO policy were evaluated directly over five independent runs, as they are deterministic with respect to training.

### 4.3.2 Average Waiting Time

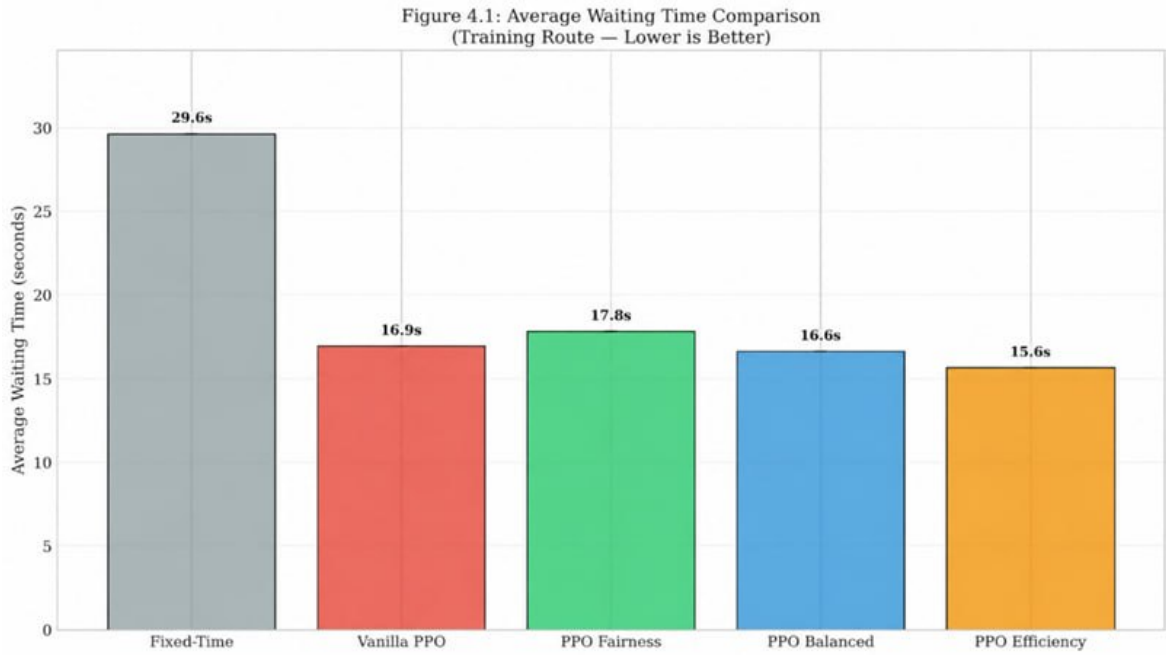


Figure 4.1: presents the average waiting time for all five controllers on the Training Route.

As shown in Figure 4.1, The results demonstrate a clear and consistent superiority of all RL-based controllers over the conventional Fixed-Time baseline. The Fixed-Time controller exhibits the highest mean waiting time (29.64 s), approximately 90% higher than that of the best-performing RL policy.

Among RL controllers, a strict performance hierarchy is observed: PPO Efficiency achieves the lowest waiting time (15.63 s), followed by PPO Balanced (16.62 s), Vanilla PPO (16.94 s), and PPO Fairness (17.82 s). Relative to Fixed-Time, these correspond to reductions of 47.3%, 43.9%, 42.8%, and 39.8%, respectively.

From a design perspective, this ordering is consistent with the underlying reward formulations. PPO Efficiency prioritises delay and throughput optimisation ( $w_1 = 0.35$ ,  $w_3 = 0.35$ ), which explains its superior waiting time performance. In contrast, PPO Fairness assigns substantial weight to directional equity ( $w_4 = 0.45$ ), introducing an explicit efficiency–fairness trade-off. Although it yields the highest waiting time among RL methods, PPO Fairness

still achieves a 39.8% reduction compared to Fixed-Time, confirming the benefit of adaptive control even under strict fairness constraints.

By contrast, PPO Fairness enforces strict equity using Jain’s Fairness Index. This constraint can over-penalise minor phase imbalances, occasionally forcing the allocation of green time to lightly loaded approaches. Such over-constrained decisions introduce a measurable efficiency loss of approximately 0.9 seconds (5.2%) compared to Vanilla PPO.

Overall, these results highlight a fundamental trade-off in multi-objective traffic signal control: efficiency-driven optimisation minimises waiting time but may compromise directional equity, whereas explicit fairness acts as a regulatory mechanism that stabilises equity at the cost of marginal reductions in operational efficiency.

### 4.3.3 Total Throughput

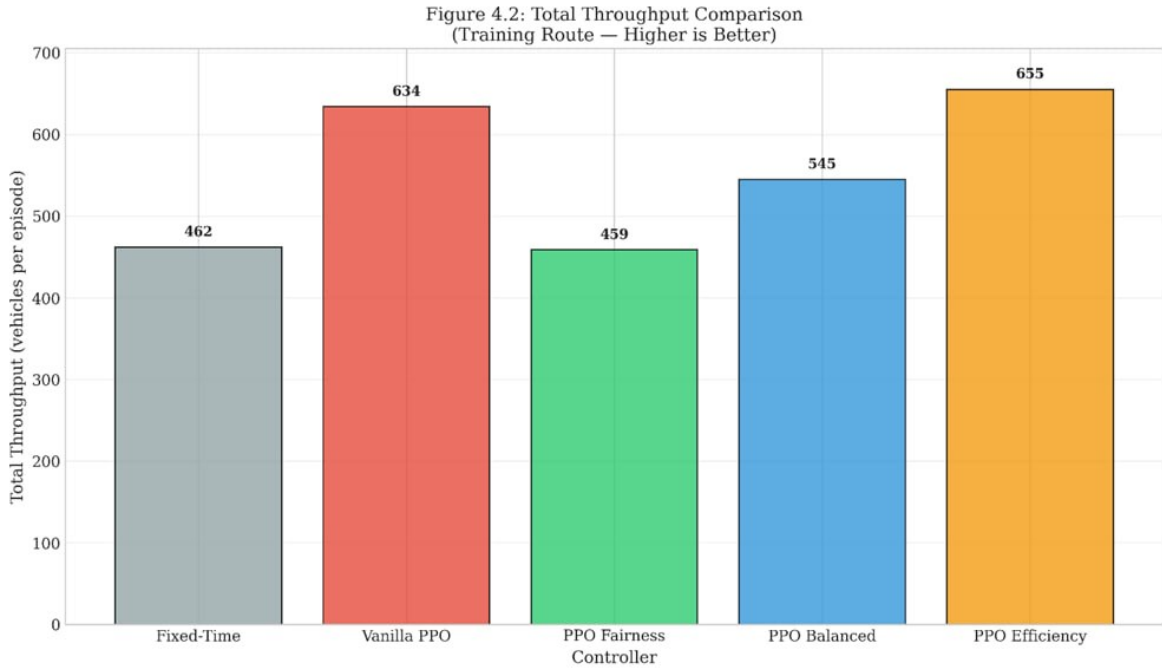


Figure 4.2: shows total throughput (vehicles per episode) on the Training Route.

As shown in Figure 4.2, The simulation results demonstrate a clear performance differentiation among the controllers regarding network throughput. PPO Efficiency achieves the highest throughput (655 vehicles), followed by Vanilla PPO (634 vehicles), PPO Balanced (545 vehicles), the Fixed-Time baseline (462 vehicles), and PPO Fairness (459 vehicles). This throughput ranking closely mirrors the underlying reward priorities of each configuration. PPO Efficiency secures the highest vehicle clearance because its reward formulation explicitly promotes vehicle discharge ( $w_3 = 0.35$ ) alongside aggregate delay reduction ( $w_1 = 0.35$ ). Conversely, PPO Fairness aggressively sacrifices vehicle clearance to maintain strict directional equity ( $w_4 = 0.45$ ), resulting in a throughput level that underperforms even the non-adaptive Fixed-Time controller (459 vs. 462 vehicles).

This severe drop indicates that a substantial portion of green time is expended on balancing East–West and North–South service delays rather than maximising vehicle progression. Meanwhile, PPO Balanced occupies an intermediate compromise position (545 vehicles), confirming that an evenly distributed objective weighting ( $w_1 = 0.30$ ,  $w_2 = 0.20$ ,  $w_3 = 0.25$ ,  $w_4 = 0.25$ ) successfully balances efficiency maximisation with fairness preservation.

Two specific operational observations merit particular attention. First, the substantial gap between PPO Efficiency (655 vehicles) and PPO Fairness (459 vehicles) represents a net difference of 196 vehicles per episode, directly quantifying the empirical cost of the efficiency–fairness trade-off in this grid network. Second, Vanilla PPO (634 vehicles) achieves a high throughput approaching that of PPO Efficiency, which is consistent with its unconstrained reward formulation that heavily prioritises delay reduction ( $w_1 = 1.00$ ) and queue minimisation ( $w_2 = 0.30$ ) without any fairness regulatory terms ( $w_4 = 0.00$ ).

By focusing strictly on traffic-flow volume, Vanilla PPO naturally favours control policies that clear highly loaded approaches aggressively. While this aggressive clearance maximises aggregate throughput, it comes at the expense of minor approaches, explaining the localized congestion and queue spillback effects that paradoxically penalise its net travel time metrics despite its high volume processing.

#### 4.3.4 Fairness Jain’s Index

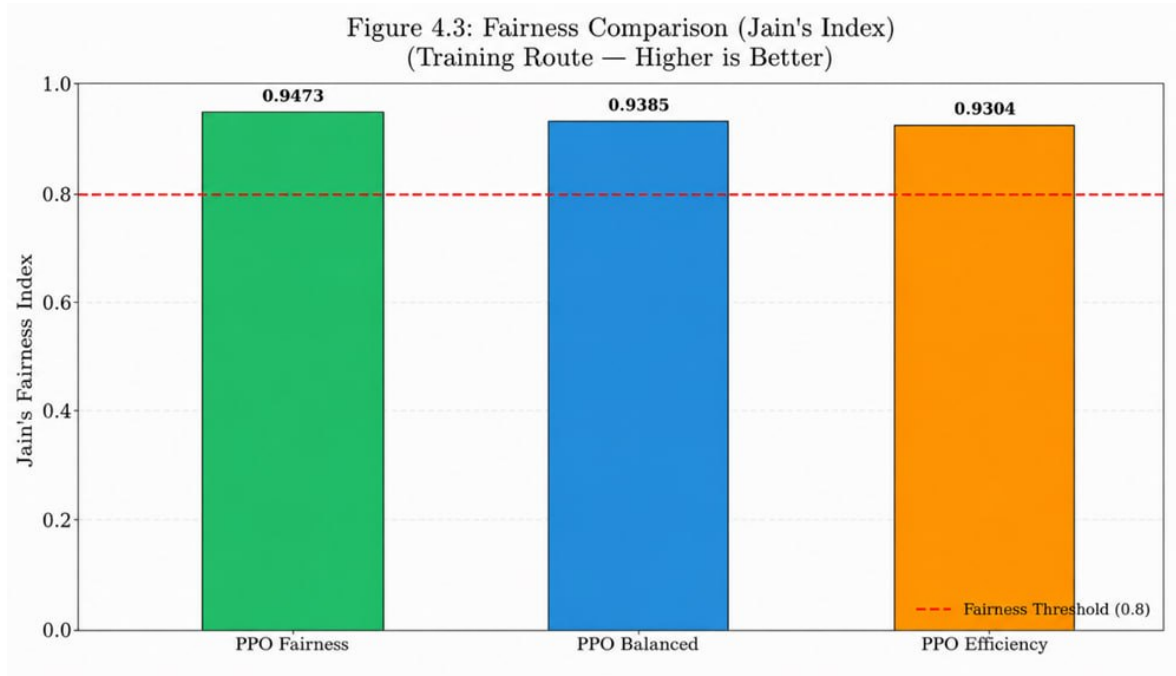


Figure 4.3: illustrates Jain’s Fairness Index for controllers configured with explicit fairness weights (*PPO Fairness*, *PPO Balanced*, and *PPO Efficiency*) on the Training Route.

**Note**

Vanilla PPO (fairness weight = 0.0) and Fixed-Time (non-adaptive) are excluded from this figure, as they lack explicit fairness objectives.

Figure 4.3 presents All three fairness-weighted controllers (PPO Fairness, PPO Balanced, and PPO Efficiency) comfortably satisfy the 0.80 fairness threshold on the Training Route.

As expected, PPO Fairness achieves the highest Jain's Index (0.9473), followed by PPO Balanced (0.9385) and PPO Efficiency (0.9304). This ordering is consistent with their respective fairness weight allocations, indicating that higher values of  $w_4$  lead to improved directional equity as measured by Jain's Index.

However, the differences between controllers are relatively small (maximum gap of 0.0169), indicating diminishing returns beyond moderate fairness weighting. This is due to the inherent trade-off between fairness and efficiency objectives, where higher fairness weights increasingly constrain optimal phase allocation, limiting additional gains in equity.

These results also confirm the relevance of the 0.80 threshold from traffic engineering literature. All controllers exceed this benchmark, indicating acceptable service equity under all fairness configurations.

### 4.3.5 Average Travel Time

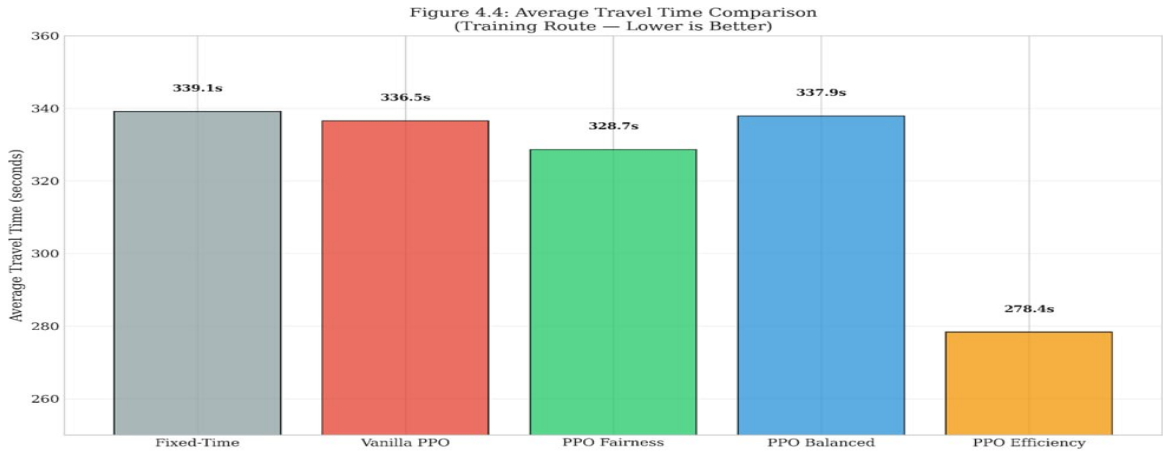


Figure 4.4: Average travel time results on the Training Route.

**Note**

The y-axis starts at 260 s to enhance visual discrimination between controllers.

As can be seen in Figure 4.4, PPO Efficiency records the lowest travel time (278.4 s), reflecting its strong prioritisation of delay minimisation ( $w_1 = 0.35$ ) and throughput maximisation ( $w_3 = 0.35$ ). This corresponds to a 17.9% reduction compared to the conventional Fixed-Time baseline (339.1 s).

The remaining controllers exhibit relatively close performance, ranging from 328.7 s to 339.1 s: PPO Fairness (328.7 s), Vanilla PPO (336.5 s), PPO Balanced (337.9 s), and Fixed-Time (339.1 s). This narrow range indicates that, under the Training Route demand structure, moderate reward configurations lead to comparable travel time outcomes.

PPO Balanced (337.9 s) performs similarly to Fixed-Time, consistent with its evenly distributed weight configuration ( $w_1 = 0.30$ ,  $w_2 = 0.20$ ,  $w_3 = 0.25$ ,  $w_4 = 0.25$ ), which avoids strong prioritisation of any single objective.

Notably, PPO Fairness achieves a lower travel time than Vanilla PPO, despite the latter lacking explicit fairness constraints ( $w_4 = 0.00$ ). This suggests that fairness-oriented phase allocation may indirectly influence network performance by reducing local congestion effects within the  $2 \times 2$  grid topology.

### 4.3.6 Average Queue Length

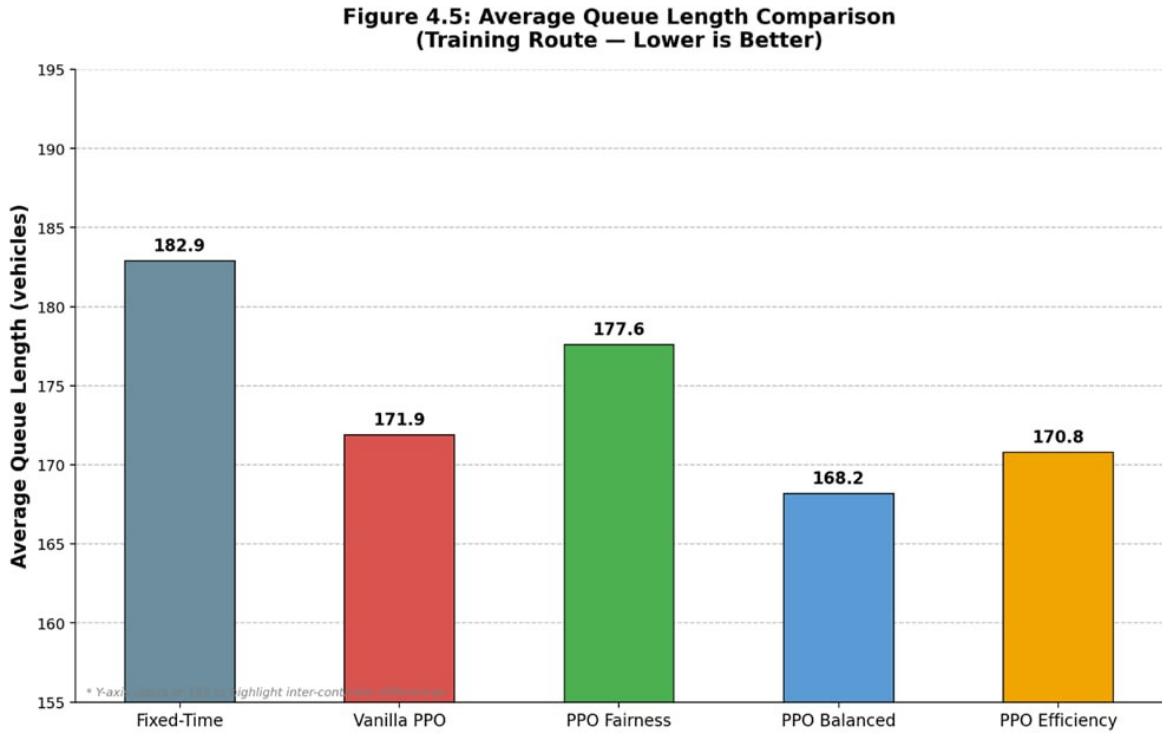


Figure 4.5: presents the average queue length across the network on the Training Route.

#### Note

The y-axis begins at 155 vehicles to highlight inter-controller differences.

As shown in Figure 4.5, The evaluation of average queue length across the network corridors reveals a consistent superiority of all learning-based frameworks over the conventional Fixed-Time baseline, with spatial footprint reductions ranging from 2.9% to 8.0%. The

Fixed-Time controller exhibits the highest network congestion, maintaining a mean queue length of 182.9 vehicles, which directly confirms its inability to adapt to transient demand fluctuations.

Intriguingly, the performance ranking changes compared to the waiting time results when evaluating queue lengths. PPO Balanced achieves the shortest average queue length (168.2 vehicles), representing an 8.0% reduction relative to Fixed-Time, despite ranking second in mean waiting time (16.62 s). This exceptional spatial performance is directly attributable to its evenly distributed reward structure ( $w_1 = 0.30, w_2 = 0.20, w_3 = 0.25, w_4 = 0.25$ ). By allocating green time more equitably across conflicting phases, PPO Balanced prevents excessive queue accumulation along any single approach.

Conversely, PPO Efficiency ranks second in queue length reduction (170.8 vehicles, a 6.6% improvement), despite capturing the global minimum in aggregate waiting time (15.63 s). This decoupling highlights a fundamental distinction between the spatial and temporal dimensions of urban traffic congestion dynamics. While queue length characterizes *spatial accumulation* (the physical volume of vehicles stored within the network infrastructure), waiting time measures *temporal delay* (the duration of vehicle immobilization). Because PPO Efficiency aggressively prioritises high-volume arterial corridors ( $w_1 = 0.35, w_3 = 0.35$ ), it successfully minimizes total temporal delay, yet occasionally permits temporary spatial queue build-ups along secondary North–South approaches.

Vanilla PPO occupies a middle tier with a mean queue length of 171.9 vehicles, consistent with its unconstrained efficiency-focused objectives. Meanwhile, PPO Fairness exhibits the most marginal improvement over the baseline, reducing queue length by only 2.9% (–5.3 vehicles). This over-constrained optimization behaviour stems from its dominant equity weight ( $w_4 = 0.45$ ), which periodically forces sub-optimal phase transitions that disrupt platoon progression and induce localized vehicle accumulation.

### 4.3.7 Episode Reward Comparison

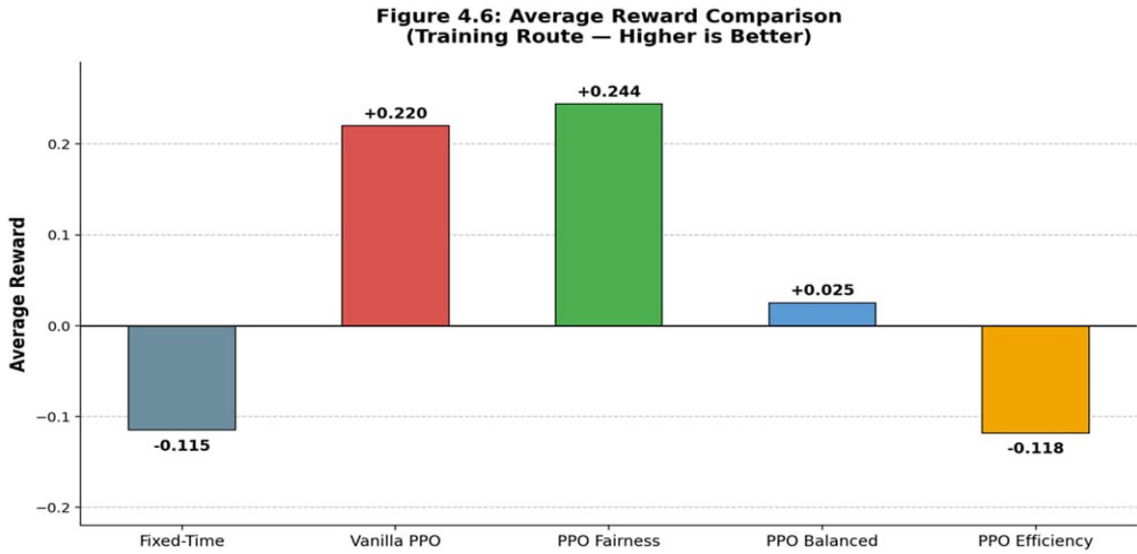


Figure 4.6: presents the mean episode reward achieved by each controller on the Training Route.

#### Importantly

these reward values are not directly comparable across PPO variants, as each optimises a distinct reward function with variant-specific weight configurations; the figure demonstrates convergence under each variant's own objective rather than cross-variant superiority.

As shown in Figure 4.6, the cumulative rewards follow a strict descending order:: PPO Fairness (+0.244), Vanilla PPO (+0.220), PPO Balanced (+0.025), Fixed-Time (−0.115), and PPO Efficiency (−0.118). PPO Fairness achieves the highest positive reward, reflecting effective optimization under its heavily equity-weighted objective function ( $w_4 = 0.45$ ). This is followed closely by Vanilla PPO, whose positive return is driven primarily by its unconstrained optimization path that heavily prioritises delay reduction ( $w_1 = 1.00$ ).

Conversely, PPO Balanced earns a modest positive reward; its distributed weight configuration ( $w_1 = 0.30, w_2 = 0.20, w_3 = 0.25, w_4 = 0.25$ ) diffuses the optimization gradient across multiple competing dimensions, yielding a weaker aggregate incentive that converges to a lower peak reward.

The lower spectrum of the reward ranking presents a compelling paradox. PPO Efficiency records the absolute lowest reward among all learning-based alternatives (−0.118), underperforming even the non-adaptive Fixed-Time baseline (−0.115), despite capturing the global optimum in real-world metrics such as throughput (655 vehicles) and waiting time (15.63 s).

This discrepancy stems directly from the structural design of the underlying reward func-

tion. From a pure traffic-engineering perspective, prioritizing the higher-demand East–West corridor is mathematically optimal for maximizing network-wide clearance. However, the formulation integrates an explicit phase imbalance penalty ( $w_4$ ) designed to penalize temporal waiting time asymmetry between conflicting approaches to prevent minor direction starvation. By aggressively favouring the dominant arterial flow, PPO Efficiency repeatedly triggers this equity penalty, suppressing its aggregate episode reward below zero.

Consequently, this negative reward represents an intentional multi-objective tension embedded within the reward shaping process rather than a failure of algorithmic convergence. This finding delivers a critical takeaway regarding reward engineering in deep reinforcement learning: a high cumulative reward signal does not guarantee superior real-world performance, nor does a negative reward imply poor operational outcomes. This divergence strongly motivates the necessity of multi-metric evaluations that extend beyond aggregate episode rewards alone.

#### 4.3.8 Directional Fairness: EW vs NS Waiting Time

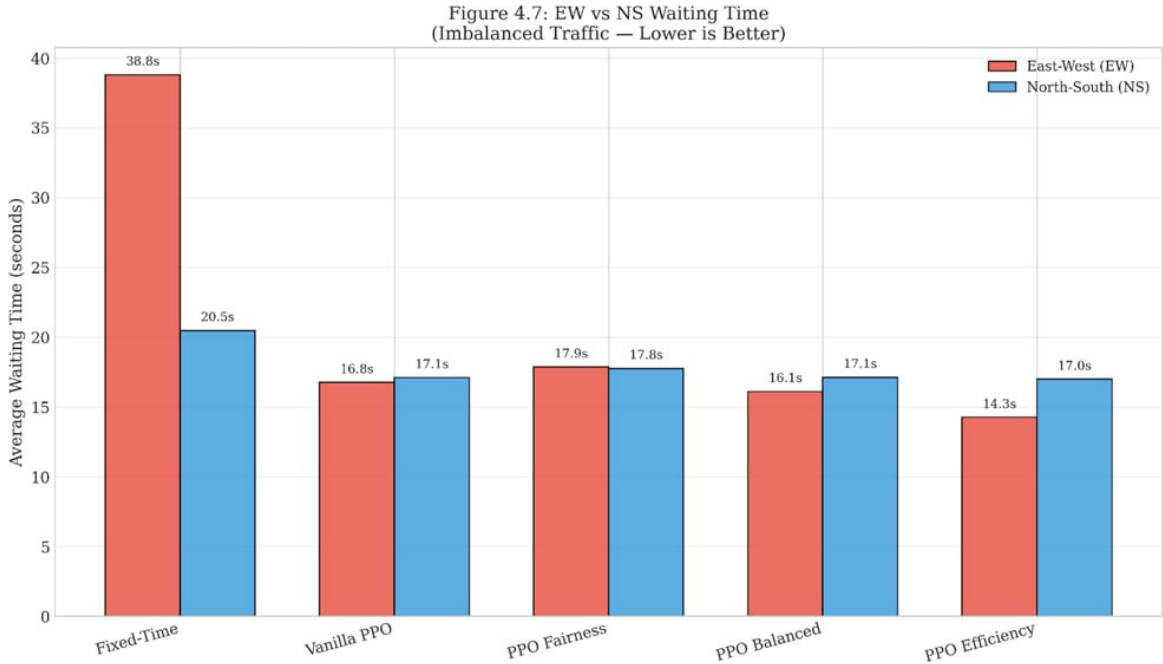


Figure 4.7: disaggregates waiting time by traffic direction (EW vs NS) under imbalanced demand conditions.

As shown in Figure 4.7, The conventional Fixed-Time controller exhibits the most severe directional imbalance, with an East–West (EW) waiting time of 38.8 s compared to 20.5 s for the North–South (NS) corridor, yielding an extreme operational gap of 18.3 s. This significant disparity reflects the fundamental limitation of static phase allocation; under highly asymmetric demand patterns, rigid green-time splits consistently over-serve lower-demand directions while starving heavier arterial flows, culminating in severe directional inequity.

Conversely, all reinforcement learning-based controllers fundamentally mitigate this imbalance, spanning a clear operational fairness spectrum dictated directly by their underlying reward formulations.

PPO Fairness achieves a near-perfect directional balance with a negligible gap of only 0.1 s (EW = 17.9 s, NS = 17.8 s), which represents a direct consequence of its heavy explicit equity weight ( $w_4 = 0.45$ ). This is followed closely by Vanilla PPO and PPO Balanced, which successfully restrict their directional gaps to 0.3 s and 1.0 s, respectively. Even PPO Efficiency, which aggressively prioritises system-wide throughput ( $w_3 = 0.35$ ) and global delay minimization ( $w_1 = 0.35$ ), bounds its directional asymmetry to a manageable 2.7 s, heavily outperforming the non-adaptive baseline.

Table 4.2 summarizes the directional data numerically, revealing systematic biases in phase allocation across all evaluated controllers under the imbalanced traffic scenario.

Table 4.2: EW vs NS Directional Waiting Time Imbalanced Traffic Scenario

| Controller     | EW Wait (s) | NS Wait (s) | Imbalance (s) | Bias Direction  |
|----------------|-------------|-------------|---------------|-----------------|
| Fixed-Time     | 38.8        | 20.5        | 18.3          | EW (extreme)    |
| Vanilla PPO    | 16.8        | 17.1        | 0.3           | NS (negligible) |
| PPO Fairness   | 17.9        | 17.8        | 0.1           | None (balanced) |
| PPO Balanced   | 16.1        | 17.1        | 1.0           | NS (low)        |
| PPO Efficiency | 14.3        | 17.0        | 2.7           | NS (high)       |

#### Note

Imbalance = |EW Wait – NS Wait|. Bias Direction indicates which direction receives systematically less green time.

While Vanilla PPO demonstrates remarkable directional equity (a 0.3 s gap) despite carrying zero explicit fairness weight ( $w_4 = 0.00$ ), this emergent behavior must be interpreted with caution. This contextual balance occurs primarily because, under the static distribution of the Training Route, optimizing global efficiency naturally aligns with maintaining directional phase balance to prevent catastrophic queue propagation. Because Vanilla PPO lacks a dedicated structural constraint, its directional equity is not mathematically guaranteed across shifted, dynamic, or unseen demand distributions.

This critical limitation justifies the necessity of fairness-oriented frameworks, such as PPO Balanced and PPO Fairness, which utilize explicit reward shaping to stabilize equity under deployment uncertainty. Ultimately, these directional results deliver a crucial insight: the transition from static to adaptive learning-based control fundamentally transforms network-wide fairness regardless of the specific reward design, while explicit weight modulation

provides the fine-grained regulatory mechanism required to calibrate the final operational trade-off.

### 4.3.9 Overall Performance Profile Radar Chart

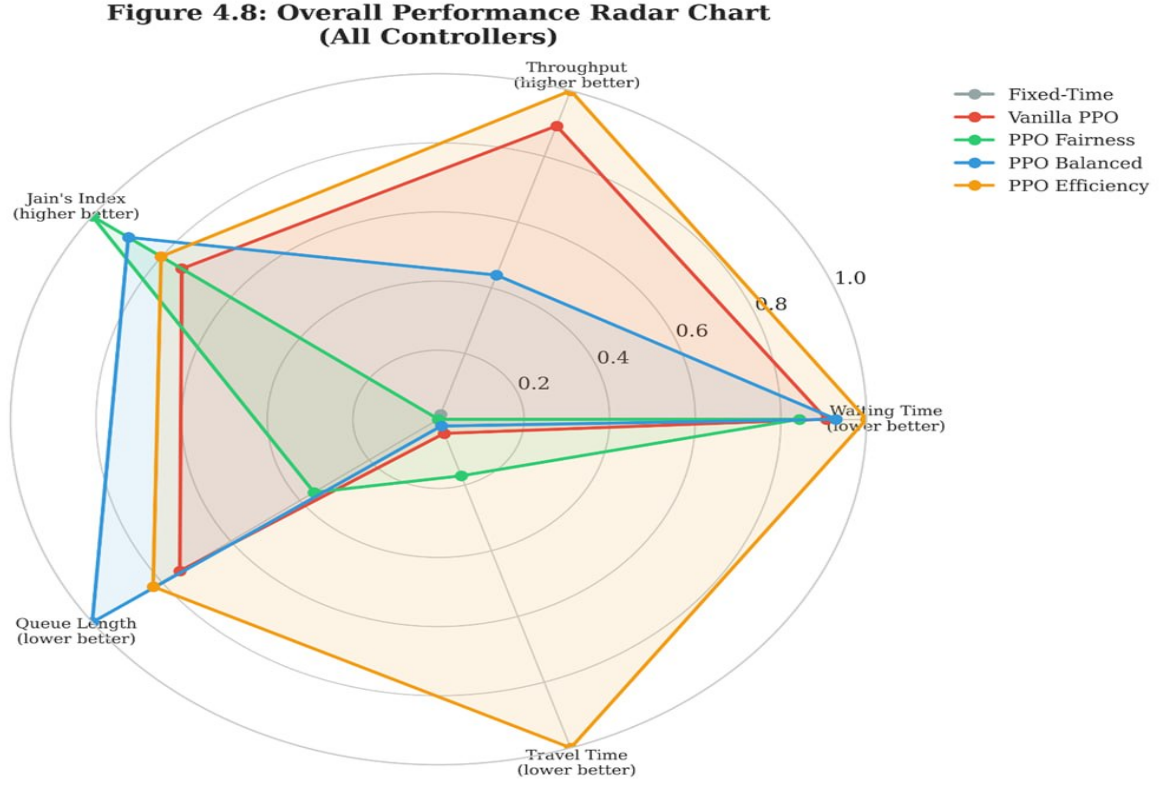


Figure 4.8: presents a normalized radar chart consolidating all five performance metrics across controllers.

All metrics are normalised to  $[0,1]$ ; As shown in Figure 4.8, lower-is-better metrics (waiting time, travel time, queue length) are inverted such that a larger polygon area indicates superior overall performance. Polygon area ranking (largest to smallest):

PPO Efficiency > Vanilla PPO > PPO Balanced > PPO Fairness > Fixed-Time, computed from normalised metric values.

**PPO Efficiency** occupies the largest polygon area, demonstrating dominant performance across throughput, waiting time, travel time, and queue length. However, its polygon contracts noticeably along the Jain's Index axis, reflecting the directional asymmetry documented in Figure 4.7, a characteristic signature of efficiency-first reward design.

**PPO Balanced** achieves the shortest queue length (168.2 vehicles) and the second-highest Jain's Fairness Index (0.9385). However, its throughput (545 vehicles) and travel time (337.9 s) are moderate compared to PPO Efficiency. This reflects its distributed weight configuration, which avoids extreme specialisation. While not the best in every metric, it is the most broadly consistent controller, preventing catastrophic failure in any single objective.

**PPO Fairness** achieves the highest Jain’s Index vertex, but its polygon collapses sharply along the throughput axis nearly overlapping with Fixed-Time (459 vs. 462 vehicles), illustrating the direct cost of fairness-oriented phase allocation on vehicle clearance rates. This trade-off is the most visually prominent feature of the chart.

**Vanilla PPO** maintains a moderately large polygon with balanced proportions, performing well in throughput and waiting time while achieving reasonable fairness as a by product of delay optimisation.

**Fixed-Time** produces the smallest polygon, collapsing toward the centre across all axes and confirming its consistently poor performance relative to all RL controllers.

#### **Key takeaway:**

No single controller dominates all axes simultaneously. The radar chart visualizes the fundamental multi-objective tension in traffic signal control:

PPO Efficiency maximizes aggregate performance at the cost of directional equity, while PPO Fairness achieves the highest equity at the cost of throughput.

PPO Balanced represents the most practical compromise for deployment scenarios requiring consistent performance across all objectives. This finding suggests that controller selection should be driven by deployment context: PPO Efficiency for throughput-critical scenarios, PPO Fairness for equity-sensitive environments, and PPO Balanced as a robust general-purpose alternative.

## **4.4 Generalisation Performance on Unseen Route**

Generalisation capability the ability of a trained policy to maintain effective performance on previously unseen traffic demand patterns without retraining is a critical requirement for any real world deployable controller. This section evaluates all five controllers on the Unseen Route (`unseen_route.xml`) in zero-shot transfer conditions.

To provide an overall assessment of the controllers’ generalisation capability, the key performance indicators obtained on both the training and unseen routes are summarised in Table 4.3. The table compares throughput (TP) and fairness (Jain’s Index), together with the performance variation between the two scenarios, highlighting the ability of each controller to maintain its effectiveness when exposed to previously unseen traffic conditions.

Table 4.3: Generalisation Performance Summary Training vs Unseen Route

| Controller     | Train TP | Unseen TP | TP Delta | Train Jain | Unseen Jain | Jain Delta |
|----------------|----------|-----------|----------|------------|-------------|------------|
| Fixed-Time     | 462      | 683       | +221     | —          | —           | —          |
| Vanilla PPO    | 634      | 646       | +12      | —          | —           | —          |
| PPO Fairness   | 459      | 600       | +141     | 0.9473     | 0.8314      | -0.1159    |
| PPO Balanced   | 545      | 559       | +14      | 0.9385     | 0.8300      | -0.1085    |
| PPO Efficiency | 655      | 560       | -95      | 0.9304     | 0.8208      | -0.1096    |

Jain’s threshold = 0.8. Dashes indicate controllers without explicit fairness weights (see Section 4.2.2).

Key observations:

- PPO Fairness retains 130.7% of its training throughput ( $459 \rightarrow 600$ ) and maintains Jain’s Index above the 0.8 threshold (0.8314). This demonstrates that fairness-oriented reward shaping (fairness weight = 0.45) produces policies that generalise robustly to unseen demand patterns.
- PPO Efficiency is the only controller with negative throughput transfer ( $655 \rightarrow 560$ ,  $\Delta = -95$ ). Its greedy reward mechanism overfitted to the training route’s permanent East-West dominance, rendering it incapable of adapting to the sudden directional inversion of the unseen layout. This confirms that aggressive specialisation for throughput on the training route comes at the cost of poor generalisation.
- Fixed-Time shows the largest throughput increase (+221, from 462 to 683). This exceptional surge directly validates the global temporal symmetry of the (`unseen_route.xml`) file. Because the static controller splits green time equally, it aligns perfectly by chance with a demand profile that distributes its heavy traffic volume equally between the North-South corridor (first 1,800 seconds) and the East-West corridor (subsequent 1,800 seconds). However, this reflects fortuitous demand alignment, not adaptive generalisation.
- Vanilla PPO and PPO Balanced show near-perfect throughput retention (101.9% and 102.6%).

### 4.4.1 Waiting Time on Unseen Route

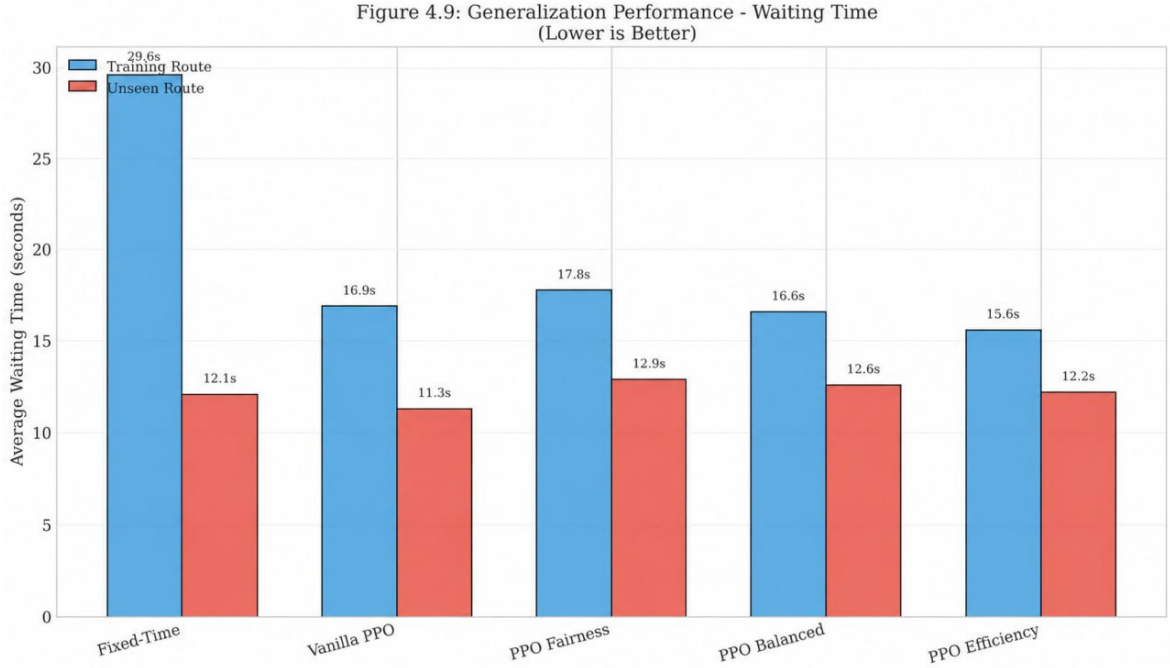


Figure 4.9: presents a side-by-side comparison of waiting times on both routes.

The figure 4.9 shows that All controllers produce lower average waiting times on the Unseen Route than on the Training Route, attributable to the unseen demand profile’s lower vehicle density.

Table 4.4: Waiting Time: Training vs Unseen Route Comparison

| Controller     | Training (s) | Unseen (s) |
|----------------|--------------|------------|
| PPO Efficiency | 15.6         | 12.2       |
| PPO Balanced   | 16.6         | 12.6       |
| Vanilla PPO    | 16.9         | 11.3       |
| PPO Fairness   | 17.8         | 12.9       |
| Fixed-Time     | 29.6         | 12.1       |

Key Findings and Generalisation Mechanisms on the Unseen Route:

**Vanilla PPO Emergence (Rank 3 → 1):** Vanilla PPO achieves the lowest absolute waiting time on the Unseen Route (11.3 s), improving its relative ranking from third to first place. This indicates that delay-focused multi-objective policies (leveraging a normalised  $\bar{w}_1 \approx 0.667$  delay weight and  $\bar{w}_2 \approx 0.200$  queue penalty) generalise exceptionally well under temporally-varying symmetric profiles. Because the model is optimised to actively suppress

queue accumulation, it dynamically capitalises on the alternating demand phases to flush out queues before downstream nodes saturate.

**PPO Efficiency Degradation (Rank 1  $\rightarrow$  3):** PPO Efficiency yields its top ranking, dropping from first to third place despite achieving the best training baseline (15.6 s). This structural shift confirms that its aggressive specialisation toward the training route’s permanent spatial imbalance induced policy crystallisation. When confronted with the dynamic directional inversions of the unseen layout, its learned control logic remains partially overfitted to historical training corridors, showing the smallest waiting time improvement (−3.4 s).

**PPO Fairness Operational Paradox:** PPO Fairness exhibits the highest average waiting time among the RL controllers (12.9 s). Its explicit mathematical constraints (fairness weight  $w_4 = 0.45$ ) force the policy to consistently prioritise cross-phase service equity, a conservative mechanism that inherently caps aggressive localised waiting time minimisation.

**Resolution of the Throughput-Delay Paradox:** Despite ranking last among RL controllers in unseen delay, PPO Fairness simultaneously achieves the strongest positive throughput transfer (+141 vehicles, retaining 130.7% of its training baseline). This non-intuitive phenomenon highlights a profound traffic engineering insight: by strictly enforcing service equity across competing phases, the controller effectively bounds maximum queue lengths on minor approaches.

This prevents catastrophic spatial spillbacks and local node blockages, keeping the overall grid open and maximising macro-level vehicle clearance (throughput) even if individual vehicles experience slightly longer micro-level holding delays to maintain the mathematical index threshold.

**Fixed-Time Fortuitous Bounce (Rank 5  $\rightarrow$  2):** The static Fixed-Time baseline shows the largest absolute delay reduction (−17.5 s), climbing from last to second place overall. As established, this behaviour is entirely decoupled from intelligent adaptation; it represents a fortuitous alignment between the rigid 125-second/125-second phase splits (25 control steps at a 5-second interval) and the unseen route’s global temporal symmetry, where total cumulative volume is evenly distributed between the corridors over the full episode duration.

**Synthesis Takeaway:** While all controllers experience lower waiting time profiles due to the temporal relaxation and alternating structure of the Unseen Route (which mitigates permanent, non-recoverable gridlocks), their relative behavioural rankings shift radically. Extreme efficiency specialisation (PPO Efficiency) proves brittle under deployment uncertainty. Conversely, multi-metric evaluation demonstrates that fairness-constrained reward shaping operates as a robust policy regulariser sacrificing minor localised delay bounds to

guarantee network-wide structural resilience and superior zero-shot throughput transfer.

4.4.2 Throughput Generalisation

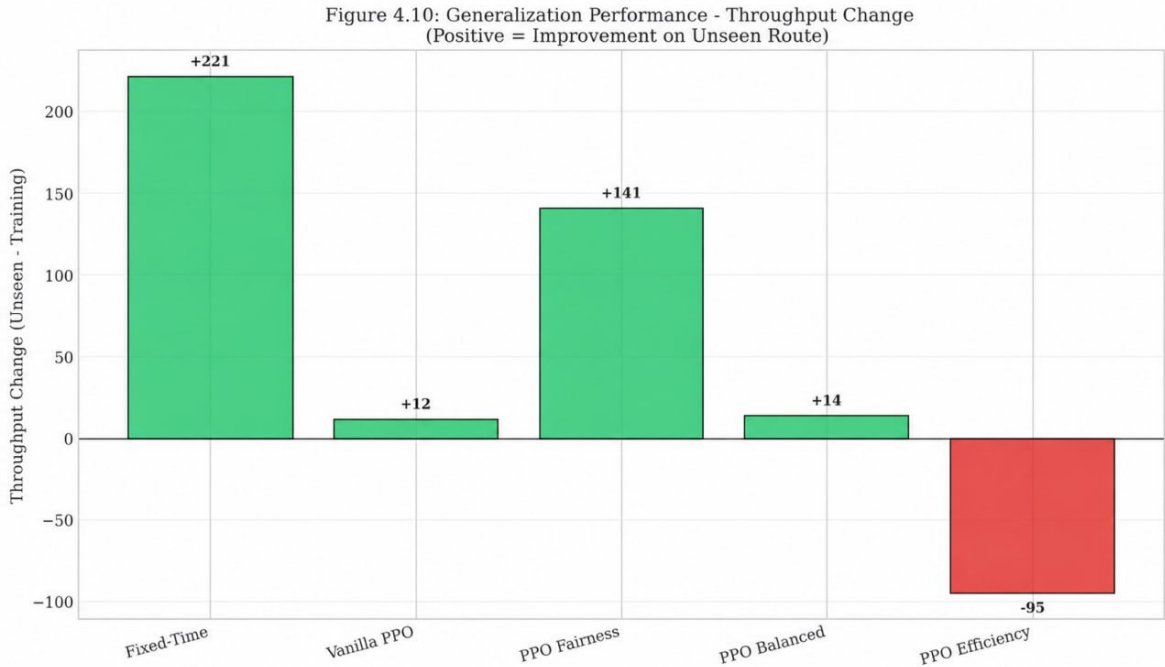


Figure 4.10: presents the absolute throughput change (Unseen TP - Training TP).

As shown in Figure 4.10, the results reveal sharp differences in the generalisation robustness of the evaluated controllers.

Table 4.5: Total Throughput (TP): Training vs Unseen Route Comparison

| Controller     | Train TP | Unseen TP | TP Δ   | Retention |
|----------------|----------|-----------|--------|-----------|
| Fixed-Time     | 462      | 683       | ↑ +221 | 147.8%    |
| PPO Fairness   | 459      | 600       | ↑ +141 | 130.7%    |
| PPO Balanced   | 545      | 559       | ↑ +14  | 102.6%    |
| Vanilla PPO    | 634      | 646       | ↑ +12  | 101.9%    |
| PPO Efficiency | 655      | 560       | ↓ -95  | 85.5%     |

- Key observations:
- Fixed-Time shows the largest absolute gain (+221), but this reflects fortuitous alignment with the unseen route’s demand profile, not adaptive generalisation.

- PPO Fairness achieves the best throughput generalisation among RL controllers (+141, 130.7% retention), confirming that fairness-oriented reward shaping produces policies that transfer robustly.
- PPO Balanced (+14, 102.6%) and Vanilla PPO (+12, 101.9%) show near-perfect throughput maintenance.
- PPO Efficiency is the only controller with negative throughput transfer (-95, 85.5% retention), the most direct evidence of route-specific overfitting.

The contrast between PPO Fairness (+141) and PPO Efficiency (-95), a swing of 236 vehicles, quantifies the generalisation cost of over-specialisation in this experimental setting.

Throughput retention ranking (RL controllers only):

**PPO Fairness (130.7%) > PPO Balanced (102.6%) > Vanilla PPO (101.9%) > PPO Efficiency (85.5%).**

### 4.4.3 Fairness Generalisation

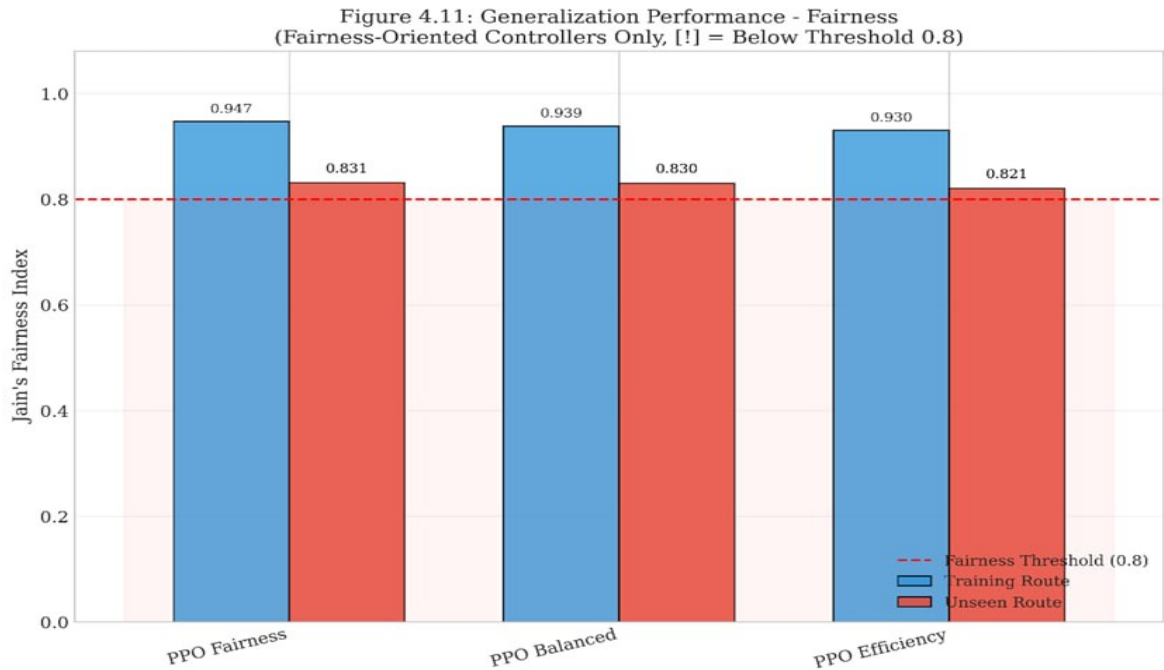


Figure 4.11: shows side-by-side Jain's Index values on Training and Unseen Routes.

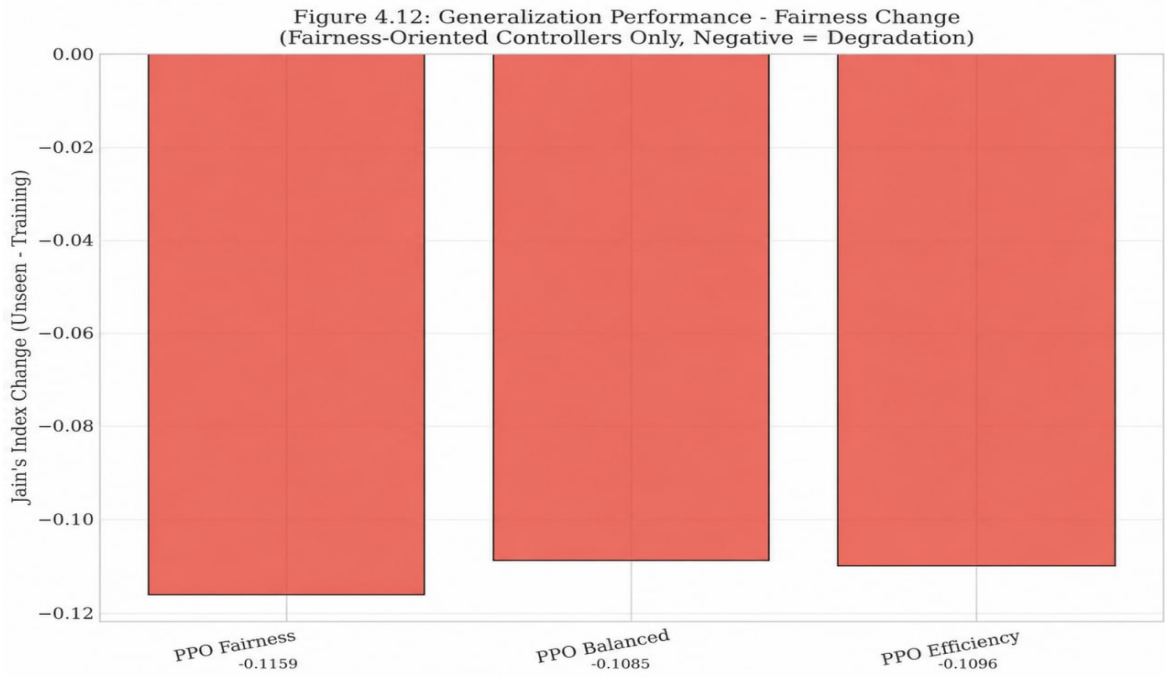


Figure 4.12: presents the absolute change (Unseen - Training).

#### Note

Vanilla PPO (fairness weight = 0.0) and Fixed-Time (non-adaptive) are excluded from this figure, as they lack explicit fairness objectives.

#### Key findings:

Table 4.6: Jain's Fairness Index: Training vs Unseen Route Comparison

| Controller     | Train Jain | Unseen Jain | $\Delta$ (Drop) | Above 0.8? |
|----------------|------------|-------------|-----------------|------------|
| PPO Fairness   | 0.9473     | 0.8314      | -0.1159         | Yes        |
| PPO Balanced   | 0.9385     | 0.8300      | -0.1085         | Yes        |
| PPO Efficiency | 0.9304     | 0.8208      | -0.1096         | Yes        |

As how the table 4.6 and the figure 4.11 shows that all three controllers maintain Jain's Index above the 0.8 threshold on the Unseen Route, demonstrating that explicit fairness weighting (fairness weight  $\geq 0.15$ ) provides robust generalization for directional equity.

**PPO Fairness** achieves the highest unseen fairness (0.8314). Although it exhibits a marginally larger absolute drop (-0.1159) compared to PPO Balanced (-0.1085) and PPO Efficiency (-0.1096), the differences are minimal ( $\approx 0.007$ ). This reflects its higher starting point (0.9473) and confirms that higher fairness weights produce not only better training fairness but also better fairness retention under distribution shift.

**PPO Balanced** (0.8300) and **PPO Efficiency** (0.8208) show very similar unseen fairness and

drop magnitudes (-0.1085 and -0.1096), suggesting that beyond a modest threshold (fairness weight  $\geq 0.15$ ), fairness generalization does not improve significantly with higher fairness weights.

**Takeaway:** Explicit fairness weighting ( $\geq 0.15$ ) ensures that controllers maintain acceptable directional equity ( $\geq 0.8$ ) even under distribution shift. However, only PPO Fairness (0.45 weight) achieves the highest unseen fairness (0.8314), while PPO Balanced and PPO Efficiency produce nearly identical generalization outcomes ( $\approx 0.825$ -0.830).

## 4.5 Discussion of Results

### 4.5.1 Efficiency-Fairness Trade-Off Analysis

The experimental results reveal a measurable efficiency–fairness trade-off across the PPO variant family. On the Training Route, transitioning from PPO Efficiency to PPO Fairness improves Jain’s Fairness Index by +0.0169 (0.9304  $\rightarrow$  0.9473), but reduces throughput by 196 vehicles (655  $\rightarrow$  459, -30.0%). This trade-off is highly asymmetric, indicating that relatively modest fairness gains may require a disproportionately large sacrifice in traffic throughput.

PPO Balanced (fairness weight = 0.25) represents a productive compromise between efficiency and equity objectives. It achieves a Jain’s Index of 0.9385, only 0.0088 below PPO Fairness, while maintaining a throughput of 545 vehicles, exceeding PPO Fairness by 86 vehicles. This result suggests that moderate fairness regularization yields a more favourable Pareto balance between throughput optimization and equitable traffic distribution.

Its distributed reward-weight configuration prevents excessive specialization toward a single objective, enabling consistently strong performance across all evaluation metrics. Although its Travel Time and Throughput remain below PPO Efficiency, PPO Balanced maintains more stable cross-metric behaviour while avoiding the severe throughput degradation observed in PPO Fairness (Figure 4.8).

Importantly, the efficiency–fairness trade-off observed on the Training Route does not transfer uniformly under distribution shift. PPO Fairness, the most fairness-constrained controller, achieves the strongest throughput transfer on the Unseen Route (+141 vehicles), whereas PPO Efficiency, the most throughput-specialized controller, exhibits the only negative transfer (-95 vehicles). This inversion corresponds to a transfer gap of 236 vehicles between the two controllers.

This performance inversion is deeply rooted in the mathematical structure of the distribution shift. While the Training Route features a static, permanent directional imbalance

favouring East-West flows, the Unseen Route introduces a temporally-varying symmetric profile. Although local directional imbalance remains severe within isolated 1,800-second phases (first NS-heavy, then EW-heavy), the cumulative global demand over the full episode duration is balanced and symmetric.

This specific topology explains why aggressive, efficiency-only reward shaping induces severe policy crystallization; PPO Efficiency overfitted to the training route’s structural asymmetry, treating the temporal inversion of the unseen layout as an unmanageable anomaly and creating artificial bottlenecks. Conversely, incorporating fairness metrics acts as an implicit policy regulariser. By penalizing phase imbalances, the fairness-constrained reward structure prevents the agent from hardcoding directional history, forcing it instead to prioritize real-time, localized queue dynamics. Consequently, the over-specialized throughput models remain brittle under deployment uncertainty, while fairness regularization unlocks robust multi-objective adaptability and superior zero-shot transfer capabilities.

#### 4.5.2 Why Fairness-Oriented Agents Generalize Better

The superior generalization performance of PPO Fairness compared to PPO Efficiency can be explained by the structural effect of fairness-oriented reward shaping. By encouraging equitable green-time allocation across competing directions, the reward function discourages persistent prioritization of heavily loaded traffic flows.

As a result, the agent is biased toward learning a more traffic-invariant phase-selection strategy rather than overfitting to the dominant traffic distribution encountered during training. Such behaviour improves policy robustness when traffic ratios shift between training and deployment environments.

This interpretation is empirically supported by the results presented in Table 4.3. PPO Fairness retains 130.7% of its training throughput ( $459 \rightarrow 600$ ) while maintaining a Jain’s Index above 0.8 (0.8314) on the Unseen Route. In contrast, PPO Efficiency experiences negative throughput transfer ( $655 \rightarrow 560$ ,  $-95$ ) despite achieving the highest training throughput.

These findings suggest that fairness-weighted reward shaping may improve not only equity outcomes, but also generalization robustness under distribution shift. From a system-design perspective, this represents a favourable outcome in which fairness objectives simultaneously contribute to improved deployment reliability and adaptive traffic control performance.

### 4.5.3 Overfitting Evidence in PPO Efficiency

The performance behaviour of PPO Efficiency provides multiple indicators consistent with overfitting to the Training Route distribution .

- **On the Training Route:** PPO Efficiency achieves the highest throughput (655 vehicles), the lowest waiting time (15.63 s), and the lowest travel time (278.4 s), indicating strong optimization for the training traffic distribution.
- **On the Unseen Route:** Its throughput decreases by 95 vehicles (14.5% regression), making it the only controller exhibiting negative throughput transfer. Its waiting-time ranking also declines from 1st to 3rd place, while its fairness performance falls below both PPO Fairness and PPO Balanced (0.8208 vs 0.8314 and 0.8300, respectively).
- **Comparative behaviour:** PPO Fairness, despite achieving the lowest training throughput among the RL-based controllers (459 vehicles), records the strongest positive throughput transfer on the Unseen Route (+141 vehicles). In contrast, PPO Efficiency exhibits the largest performance degradation (−95 vehicles).

Taken together, these results suggest that aggressive optimization toward throughput on the Training Route may have encouraged policy specialization to route-specific traffic patterns, reducing robustness under unseen traffic conditions. The inverse relationship between peak training performance and transfer performance therefore provides evidence consistent with overfitting in PPO Efficiency.

It is important to reconcile two seemingly contradictory observations about PPO Efficiency. On one hand, Section 4.3.1 reports that PPO Efficiency receives a negative episode reward (−0.118) due to the imbalance penalty ( $\lambda = 0.15$ ) penalizing its directional bias toward EW. On the other hand, Section 4.6.1 attributes its overfitting to aggressive throughput specialization.

These observations are not contradictory; rather, they reflect a trade-off embedded in the reward function. Despite the imbalance penalty, the combined weights for throughput ( $w_3 = 0.35$ ) and delay ( $w_1 = 0.35$ ) are sufficiently high to drive the agent toward prioritizing the heavier EW direction. The penalty reduces the magnitude of the episode reward (making it negative) but does not reverse the learned preference. Consequently, the agent develops a policy that is specialised to the training route’s permanent EW dominance. When this directional pattern shifts on the Unseen Route, the policy fails to adapt, resulting in negative throughput transfer (−95 vehicles). Thus, the negative reward and the overfitting are two manifestations of the same underlying cause: aggressive efficiency optimization in an imbalanced environment.

#### 4.5.4 Correlation Between Reward and Real Traffic Metrics

A recurring observation across the PPO variants is the weak correlation between episode reward and objective-level traffic performance. This indicates a degree of reward–objective misalignment within the multi-component reinforcement learning framework.

PPO Efficiency records the lowest episode reward among the RL-based controllers ( $-0.118$ ), yet achieves the best Training Route performance in terms of throughput (655 vehicles), waiting time (15.63 s), and travel time (278.4 s). In contrast, PPO Fairness achieves the highest episode reward ( $+0.244$ ) while producing the lowest throughput among the PPO variants (459 vehicles).

These dissociations arise from the structure of the composite reward function, which combines multiple weighted components including delay, queue length, throughput, fairness, and imbalance penalties. In such settings, optimization of the aggregated reward does not necessarily translate into optimal performance for any single operational metric.

For PPO Efficiency, the negative reward is primarily associated with its recurrent imbalance penalty ( $\lambda = 0.15$ ), reflecting directional traffic bias rather than poor traffic-flow efficiency. This bias is operationally justified, as prioritizing the higher-demand EW direction maximizes throughput and minimizes waiting time under imbalanced traffic conditions. The imbalance penalty, however, discourages such behaviour as part of a deliberate design trade-off between efficiency and equity objectives.

Consequently, the reward function penalizes behaviour that remains highly effective from a throughput and travel-time perspective, illustrating the tension between fairness-oriented optimization and traffic-efficiency maximization.

This finding highlights an important methodological implication for RL-based traffic signal control. Episode reward should not be treated as a sufficient standalone indicator of deployment readiness. Instead, evaluation requires a multi-metric assessment framework incorporating throughput, waiting time, fairness, and travel time independently, ensuring that reward optimization aligns with real-world traffic-management objectives.

#### 4.5.5 The Fixed-Time Baseline Unexpected Unseen-Route Throughput

The experimental evaluation on the Unseen Route reveals an exceptional throughput surge for the Fixed-Time baseline, which spikes by +221 vehicles (moving from a training throughput of 462 to an unseen throughput of 683). This phenomenon represents a critical analytical point: it does not indicate adaptive operational generalization or real-time responsiveness

from the static controller, but rather illustrates a fortuitous global demand alignment inherent to the scenario’s temporal topology.

Crucially, as defined in the experimental setup, the Unseen Route maintains the exact same total traffic volume as the Training Route, thereby preserving dense traffic conditions across the entire 5,400-second episode. However, while the Training Route exhibits a permanently asymmetric demand pattern (favouring the East-West corridor at a 3:1 ratio), the Unseen Route introduces a temporally-varying symmetric profile. In this unseen configuration, heavy demand dominates the North-South corridor during the first 1,800 seconds and shifts symmetrically to dominate the East-West corridor during the subsequent 1,800 seconds.

Consequently, because the Fixed-Time controller employs rigid, equal green-time splits across its phases, its static operation aligns perfectly by chance with a profile whose global cumulative volume is evenly and symmetrically distributed over the full duration of the simulation. This perfect geometric alignment prevents the early formation of catastrophic spillbacks that typically choke the non-adaptive baseline under permanent spatial imbalances. This result demonstrates that static controllers can fortuitously excel when network demand distribution matches their underlying pre-timed structural symmetry, emphasising the vital importance of utilising asymmetric unseen testing profiles in future benchmarks to isolate true reinforcement learning adaptability from accidental pattern matching.

#### 4.5.6 Practical Deployment Recommendations

Based on the experimental findings, Table 4.7 maps operational priorities to the most appropriate controller configuration.

Table 4.7: Practical Controller Deployment Recommendations

| Priority                             | Recommended Controller | Key Strength                  | Caution                      |
|--------------------------------------|------------------------|-------------------------------|------------------------------|
| Unknown environment / Generalization | PPO Fairness           | Best unseen fairness (0.8314) | Lower training TP (459)      |
| Stability / Consistent performance   | PPO Balanced           | Lowest queue: 168.2 veh       | Moderate TP (545)            |
| Known environment / Max throughput   | PPO Efficiency         | Highest TP: 655 veh           | Over-fits; avoids new routes |
| No training feasible                 | Fixed-Time             | Zero complexity               | Poor direction equity        |
| PPO without fairness-aware reward    | Vanilla PPO            | Strong on familiar routes     | No fairness guarantees       |

Recommendations derived from Training Route and Unseen Route evaluation results. Jain's threshold = 0.8.

Overall, the results suggest that no single controller dominates across all operational regimes. Instead, performance is inherently context-dependent, requiring selection based on the trade-off between efficiency, fairness, and generalisation robustness.

### 4.5.7 Summary of Key Findings

1. **PPO Efficiency dominates training-route metrics:** It achieves the best waiting time (15.63 s), highest throughput (655 vehicles), and lowest travel time (278.4 s). However, it exhibits evidence consistent with overfitting, as indicated by a 95 vehicle throughput regression on the Unseen Route, making it the only controller exhibiting negative transfer.
2. **PPO Fairness demonstrates superior generalization:** It achieves the highest unseen fairness among RL controllers (0.8314), maintaining Jain's Index well above the 0.8 threshold. It also achieves the strongest positive throughput transfer among RL-based methods (+141 vehicles), despite lower training throughput (459 vehicles).
3. **PPO Balanced provides the most consistent multi-objective behaviour:** It achieves the lowest queue length (168.2 vehicles), stable cross-route performance (TP Delta = +14), and the most balanced multi-metric profile across evaluation dimensions.
4. **Episode reward is not a reliable predictor of deployment performance:** PPO Efficiency attains the lowest episode reward (−0.118) while achieving the best training-route metrics, whereas PPO Fairness achieves the highest reward (+0.244) despite lower training throughput, indicating a weak coupling between reward optimization and deployment-level performance.
5. **Fairness-oriented reward shaping improves generalization:** PPO Fairness retains 130.7% of training throughput (459 → 600) while maintaining acceptable fairness (0.8314) on the Unseen Route. In contrast, PPO Efficiency suffers negative transfer (−95 vehicles), suggesting that fairness constraints implicitly regularize the policy. This improves cross-domain robustness, but reduces peak training performance.
6. **Practical deployment recommendations:** The observed trade-offs suggest the following deployment guidelines:
  - PPO Efficiency is recommended for throughput-critical applications (e.g., highways, known demand patterns).
  - PPO Fairness is recommended for equity-sensitive applications (e.g., residential areas, unknown demand patterns).

- PPO Balanced is recommended as a general-purpose default for applications requiring consistent multi-objective performance.

(See Table 4.7 for full mapping of operational priorities to controller selection.)

## 4.6 Limitations of the Study

1. Single network topology ( $2 \times 2$  grid only): The study is conducted on a small-scale  $2 \times 2$  signalised grid network. As a result, the findings may not directly generalize to larger or more irregular urban network topologies.
2. Fixed and limited demand patterns: Training and generalisation were each evaluated using a single demand configuration, which may limit temporal diversity and real-world traffic variability.
3. Single unseen scenario for generalisation assessment: Generalisation was evaluated using only one alternative demand file, providing a single evaluation scenario and limiting statistical robustness across heterogeneous traffic conditions.
4. Simulation-based evaluation without real-world validation: All experiments are conducted within the SUMO simulation environment. Consequently, sim-to-real transfer performance remains unvalidated in real-world traffic systems.
5. Single-agent, single-network control assumption: The study assumes a single-agent control paradigm. Multi-agent or hierarchical reinforcement learning architectures may be necessary to capture inter-junction dependencies in larger-scale deployments.
6. Limited evaluation sample and seed selection bias: Although each PPO variant was trained using five random seeds, only the best-performing seed per variant was selected for final evaluation. This may introduce optimistic bias and does not fully capture performance variability across different initialisations.

## 4.7 Summary

This chapter presented a comprehensive experimental evaluation of five traffic signal controllers Fixed-Time, Vanilla PPO, PPO Fairness, PPO Balanced, and PPO Efficiency across six performance metrics on both a familiar Training Route and a novel Unseen Route. The evaluation was supported by 13 figures (Figures 4.1- 4.13) and four summary tables.

On the Training Route, the results reveal a pronounced efficiency–fairness trade-off: PPO Efficiency achieves the best throughput (655 vehicles), lowest waiting time (15.63 s), and

lowest travel time (278.4 s), while PPO Fairness achieves the highest Jain’s Fairness Index (0.9473). PPO Balanced offers the most stable multi-objective performance, achieving the lowest queue length (168.2 vehicles) and a consistently balanced profile across all metrics. All RL controllers reduce waiting time by 40–47% relative to Fixed-Time.

On the Unseen Route, the generalization analysis reveals a clear performance divergence: PPO Efficiency suffers a throughput regression of 95 vehicles, making it the only controller exhibiting negative transfer. In contrast, PPO Fairness achieves the strongest positive throughput transfer among RL controllers (+141 vehicles), retaining 130.7% of its training throughput while maintaining the highest unseen fairness (0.8314), well above the 0.8 threshold.

Notably, all fairness-weighted controllers (PPO Fairness, PPO Balanced, and PPO Efficiency) maintain Jain’s Index  $\geq 0.8$  under distribution shift, confirming that fairness weighting ( $\geq 0.15$ ) ensures acceptable equity in unseen scenarios.

These findings lead to two key conclusions:

First, fairness-oriented reward shaping simultaneously improves directional equity and generalization robustness, demonstrating a positive-sum relationship between fairness and adaptability. PPO Fairness in particular illustrates that fairness-constrained policies generalize more effectively to unseen demand patterns.

Second, efficiency-oriented reward shaping, while optimal on training-route performance, is more vulnerable to route-specific overfitting, leading to degraded performance under distribution shift. PPO Efficiency’s negative transfer (−95 vehicles) provides clear evidence of this effect. From a deployment perspective: PPO Fairness is the most reliable choice under uncertain or time-varying demand, offering the strongest generalization performance. PPO Efficiency is best suited for stable, well-characterized environments where maximum throughput is the primary objective. PPO Balanced provides the most consistent multi-objective behaviour, making it a suitable general-purpose default policy.

# Chapter 5

## Conclusion and Future Work

### 5.1 Conclusion

This thesis investigated the design, implementation, and evaluation of a reinforcement learning-based adaptive traffic signal control system using Proximal Policy Optimization (PPO). The proposed framework integrates the SUMO microscopic traffic simulator with a multi-objective reward function that jointly optimises traffic efficiency and directional fairness. Five controllers were evaluated on both a Training Route and a structurally distinct Unseen Route using six performance metrics: average waiting time, queue length, throughput, travel time, Jain’s Fairness Index, and episode reward.

The study successfully achieved all research objectives by developing a fairness-aware RL framework with three reward-weight configurations representing efficiency-, fairness-, and balanced-oriented strategies. Compared with the Fixed-Time baseline, all PPO-based controllers significantly reduced average waiting time by 39.8% to 47.3%, confirming the superiority of adaptive RL-based control under imbalanced traffic demand.

The results reveal a clear efficiency–fairness trade-off. Increasing fairness from PPO Efficiency to PPO Fairness improved Jain’s Fairness Index by only +0.0169 but reduced throughput by 196 vehicles per episode (−30%). PPO Balanced achieved the best compromise, maintaining a Jain’s Index of 0.9385 while exceeding PPO Fairness in throughput by 86 vehicles. Zero-shot evaluation on the Unseen Route further showed that PPO Fairness achieved the strongest positive transfer (+141 vehicles, 130.7% retention) and maintained directional equity (Jain = 0.8314), whereas PPO Efficiency exhibited negative transfer (−95 vehicles), indicating route-specific overfitting. Furthermore, PPO Efficiency produced the best traffic performance despite recording the lowest episode reward (−0.118), demonstrating that reward optimisation alone is insufficient and that multi-metric evaluation is essential.

Three main conclusions emerge from this work. First, fairness-oriented reward shaping not only improves directional equity but also enhances generalisation to unseen traffic conditions, revealing a positive relationship between fairness and adaptability. Second, efficiency-oriented optimisation delivers the highest training performance but increases sensitivity to distribution shifts. Third, PPO Balanced represents the most practical deployment strategy, providing stable performance across all evaluation metrics without excessive specialisation.

Finally, the proposed framework has important implications for intelligent transportation systems and smart cities. It can replace or complement conventional fixed-time controllers, reducing vehicle waiting times by over 40% while ensuring equitable service for different traffic flows. Moreover, its strong generalisation capability supports deployment under varying traffic conditions without continuous retraining, while reduced waiting times and queue lengths contribute to lower fuel consumption and carbon emissions, supporting sustainable urban mobility.

## 5.2 Future Work

Several directions can extend and strengthen the contributions of this study:

1. **Scaling to larger and more complex network topologies:** The current study is limited to a  $2 \times 2$  signalized grid network. Future work should evaluate the proposed framework on larger urban networks ( $3 \times 3$ ,  $4 \times 4$ , or real-world road networks extracted from OpenStreetMap) to assess scalability and the impact of network topology on policy generalization.
2. **Multi-Agent Reinforcement Learning (MARL):** The single-agent paradigm employed in this study controls all intersections through a centralized policy, which does not capture inter-junction dependencies and coordination dynamics. Extending the framework to a multi-agent setting where each intersection is governed by an independent or co-operatively trained agent would more accurately reflect the distributed nature of urban traffic signal control and enable coordination across adjacent intersections.
3. **Real-world validation and sim-to-real transfer:** All experiments were conducted within the SUMO simulation environment. Deploying the trained policies on real traffic signal hardware or a hardware-in-the-loop test bed is a critical next step to validate sim-to-real transfer performance and identify domain gaps between simulated and physical traffic dynamics.
4. **Dynamic and temporally varying reward configurations:** The current reward weights are fixed throughout training. Future work could explore adaptive reward shaping mechanisms that dynamically adjust efficiency–fairness weights based on real-time

traffic conditions, time of day, or network congestion levels, enabling more context-aware signal control.

5. **Incorporation of additional traffic modes:** The current framework models private vehicle flows only. Extending the state space and reward function to explicitly account for public transit (buses, trams), cyclists, and pedestrian crossings would produce a more comprehensive multi-modal traffic signal controller aligned with sustainable urban mobility goals.
6. **Longitudinal generalization across diverse demand scenarios:** Generalization was evaluated using a single unseen demand file. Future studies should assess policy robustness across a diverse set of demand scenarios including morning and evening peak hours, weekend patterns, incident-induced demand shifts, and seasonal variation to provide statistically robust generalization estimates.
7. **Integration with real time sensor data:** The current system relies on simulated TraCI observations. Integrating the framework with real world sensor inputs including loop detectors, camera based vehicle counting, and connected vehicle data would constitute a meaningful step toward operational deployment in live urban environments.

# Bibliography

- [1] Y. K. Thakre and P. Y. Pawade, “Traffic Congestion at Urban Road – Review,” *IOP Conference Series: Earth and Environmental Science*, vol. 1326, no. 1, Art. no. 012094, 2024, doi: 10.1088/1755-1315/1326/1/012094.
- [2] D. Schrank, K. Jha, L. Albert, and B. Eisele, “Urban Mobility Report 2024,” Texas A&M Transportation Institute (TTI), College Station, TX, USA, 2024.
- [3] Texas A&M Transportation Institute, “Traffic Hits Record High as Commuters Rewrite the Rush Hour,” TTI News Release, 2025.
- [4] R. E. Kadakath, M. R. Setty, C. Nanjundaswamy, A. Tantri, and S. R. Naganna, “Urban transportation challenges: Analysis and mitigation strategies for road accidents, noise pollution and environmental impacts,” *Open Engineering*, vol. 15, no. 1, Art. no. 20250123, 2025, doi: 10.1515/eng-2025-0123.
- [5] J. N. Fadila, N. H. A. Wahab, A. Alshammari, A. Aqarni, A. Al-Dhaqm, and N. Aziz, “Comprehensive Review of Smart Urban Traffic Management in the Context of the Fourth Industrial Revolution,” *IEEE Access*, vol. 12, pp. 196866–196886, 2024, doi: 10.1109/ACCESS.2024.3509572.
- [6] M. Guo, P. Wang, C.-Y. Chan, and S. Askary, “A Reinforcement Learning Approach for Intelligent Traffic Signal Control at Urban Intersections,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4242–4247, doi: 10.1109/ITSC.2019.8917268.
- [7] R. H. Al-Rubaei and I. T. Hameed, “Performance analysis, evaluation, and improvement of selected unsignalized intersection using SIDRA software – Case study,” *Open Engineering*, vol. 14, no. 1, Art. no. 20220537, 2024, doi: 10.1515/eng-2022-0537.
- [8] S. Yuan, X. Zhao, and Y. An, “Identification and optimization of traffic bottleneck with signal timing,” *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 1, no. 5, pp. 353–361, 2014, doi: 10.1016/S2095-7564(15)30281-6.
- [9] A. Kekuda, R. Anirudh, and M. Krishnan, “Reinforcement learning based intelligent traffic signal control using n-step SARSA,” in *2021 International Conference on Ar-*

- tificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 379–384, doi: 10.1109/ICAIS50930.2021.9395942.
- [10] S. Yan, J. Zhang, D. Büscher, and W. Burgard, “Efficiency and equity are both essential: A generalized traffic signal controller with deep reinforcement learning,” in *Proc. 2020 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2020, pp. 5526–5533, doi: 10.1109/IROS45743.2020.9340784.
- [11] K. F. Chu, A. Y. S. Lam, and V. O. K. Li, “Traffic Signal Control Using End-to-End Off-Policy Deep Reinforcement Learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7184–7195, 2021, doi: 10.1109/TITS.2021.3067057.
- [12] H. Wei, G. Zheng, V. Gayah, and Z. Li, “Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation,” *ACM SIGKDD Explorations Newsletter*, vol. 22, no. 2, pp. 12–18, 2021, doi: 10.1145/3447556.3447565.
- [13] Ingook Jang, Donghun Kim, Donghun Lee, and Young-Sung Son, “An Agent-Based Simulation Modeling with Deep Reinforcement Learning for Smart Traffic Signal Control,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 1028–1030, doi: 10.1109/ICTC.2018.8539377.
- [14] National Academies of Sciences, Engineering, and Medicine, “Signal Timing Manual – Second Edition,” Transportation Research Board (TRB), NCHRP Report 812, Washington, DC, USA, 2015, doi: 10.17226/22097.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., Cambridge, MA, USA: MIT Press, 2018.
- [16] F. V. Webster, “Traffic signal settings,” Road Research Laboratory (RRL), Technical Paper No. 39, London, U.K., 1958.
- [17] B. Abdulhai, R. Pringle, and G. J. Karakoulas, “Reinforcement Learning for True Adaptive Traffic Signal Control,” *Journal of Transportation Engineering (ASCE)*, vol. 129, no. 3, pp. 278–285, 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:3(278).
- [18] X. Zhang, L. S. Chan, N. Nassir, and M. Sarvi, “Towards fair lights: A multi-agent masked deep reinforcement learning for efficient corridor-level traffic signal control,” *Communications in Transportation Research*, vol. 5, 2025, Art. no. 100203, doi: 10.1016/j.commtr.2025.100203.
- [19] Li Li, Yisheng Lv, and Fei-Yue Wang, “Traffic Signal Timing via Deep Reinforcement Learning,” *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016, doi: 10.1109/JAS.2016.7508798.

- [20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” arXiv preprint arXiv:1707.06347, 2017, doi: 10.48550/arXiv.1707.06347.
- [21] W. Fang, X. Zhao, and C. Zhang, “Fairness-aware Multi-Agent Reinforcement Learning and Visual Perception for Adaptive Traffic Signal Control,” *Optoelectronics Letters*, vol. 20, no. 12, pp. 764–768, 2024, doi: 10.1007/s11801-024-3267-2.
- [22] X. Du, Z. Li, C. Long, Y. Xing, P. S. Yu, and H. Chen, “FELight: Fairness-Aware Traffic Signal Control via Sample-Efficient Reinforcement Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 9, pp. 4678–4692, 2024, doi: 10.1109/TKDE.2024.3376745.
- [23] N. Bhawe, A. Dhagavkar, K. Dhande, M. Bana, and J. Joshi, “Smart Signal - Adaptive Traffic Signal Control using Reinforcement Learning and Object Detection,” in *Proc. 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 2019, pp. 624–628, doi: 10.1109/I-SMAC47947.2019.9032589.
- [24] C. Wu, Z. Ma, and I. Kim, “Multi-agent reinforcement learning for traffic signal control: algorithms and robustness analysis,” in *Proc. IEEE ITSC*, 2020, doi: 10.1109/ITSC45102.2020.9294623.
- [25] A. Lopez, M. Behrisch, L. Bieker-Walz, et al., “Microscopic traffic simulation using SUMO,” in *Proc. IEEE Intelligent Transportation Systems Conference (ITSC)*, 2018, pp. 2575–2582, doi: 10.1109/ITSC.2018.8569938.
- [26] L. Zhu, P. Peng, Z. Lu, and Y. Tian, “MTLight: Efficient Multi-Task Reinforcement Learning for Traffic Signal Control,” arXiv preprint arXiv:2404.00886, 2024, doi: 10.48550/arXiv.2404.00886.
- [27] T. Chu, J. Wang, L. Codecà, and Z. Li, “Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019, doi: 10.1109/TITS.2019.2901791.
- [28] H. Zhang, S. Feng, C. Liu, Y. H. Son, et al., “CityFlow: A Multi-Agent Reinforcement Learning Environment for Traffic Signal Control,” in *Proc. The World Wide Web Conference (WWW)*, 2019, pp. 3620–3624, doi: 10.1145/3308558.3313441..
- [29] R. Lowrie, “SCATS: The Sydney Co-ordinated Adaptive Traffic System—Principles, Methodology, Algorithms,” in *Proc. International Conference on Road Traffic Signalling*, London, UK, 1982, pp. 67–70.

- [30] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, “The SCOOT On-Line Traffic Signal Optimisation Technique,” *Traffic Engineering & Control*, vol. 23, no. 4, pp. 190–192, 1982.
- [31] PTV Group, “PTV VISSIM User Manual and Technical Documentation,” PTV AG, Karlsruhe, Germany, 2023.
- [32] TSS Transport Simulation Systems, “AIMSUN Next User Manual and Modelling Guide,” Aimsun SLU, Barcelona, Spain, 2023.
- [33] E. Mikea, “The Power of PPO: How Proximal Policy Optimization Solves a Range of RL Problems,” *Medium*, 2024. [Online]. Available: <https://medium.com/%40emik ea03/the-power-of-ppo-how-proximal-policy-optimization-solves-a-range-of-rl-problems-10076d9da34e>
- [34] D. Byrne, “PPO Explained: Architectural Concepts and Implementation,” *GitHub Pages*, 2023. [Online]. Available: <https://djbyrne.github.io/2023/04/10/ppo-explained.html> (Accessed: May 12, 2026).
- [35] CSDN Blog, “Reinforcement Learning for Intelligent Transportation Systems,” *CSDN IT Platform*, 2026. [Online]. Available: [https://blog.csdn.net/qq\\_52944715](https://blog.csdn.net/qq_52944715)
- [36] W. Genders and S. Razavi, “Asynchronous n-step Q-learning Adaptive Traffic Signal Control,” *Journal of Intelligent Transportation Systems*, vol. 23, no. 4, pp. 319–331, 2019. doi: 10.1080/15472450.2018.1491003.
- [37] M. Towers, J. K. Terry, A. Kwiatkowski, et al., “Gymnasium: A Standard Interface for Reinforcement Learning Environments,” arXiv preprint arXiv:2407.17032, 2024, doi: 10.48550/arXiv.2407.17032.



## شهادة الترخيص بالإيداع

أنا الممضي أسفله الأستاذ: بن قنان مسعود

بصفتي مشرف- والمسؤول عن تصحيح مذكرة تخرج ماستر الموسومة بـ

**Adaptive Traffic Signal Control using Reinforcement Learning and Simulation**

من انجاز الطالبة: مهاية فاطمة والطالبة: عبد الهادي إيمان.

الكلية: العلوم والتكنولوجيا.

القسم: الرياضيات والإعلام الآلي.

الشعبة: إعلام الي.

التخصص: الأنظمة الذكية لاستخراج المعارف.

تاريخ التقييم/المناقشة: 2026/06/21

أشهد ان الطلبة قاموا بالتعديلات والتصحيحات المطلوبة من طرف لجنة المناقشة وان المطابقة بين النسخة الورقية والالكترونية استوفت جميع شروطها.

مصادقة رئيس القسم

رئيس قسم الرياضيات والإعلام الآلي  
الحاج موسى



امضاء المسؤول عن التصحيح

Benguennane Messaoud