# Targeted Knowledge Transfer for Learning Traffic Signal Plans

Nan Xu[1*], Guanjie Zheng[2], Kai Xu[3], Yanmin Zhu[1], Zhenhui Li[2]

[1] Shanghai Jiao Tong University
{xunannancy,yzhu}@sjtu.edu.cn
[2] Pennsylvania State University
{gjz5038,jessieli}@psu.edu
[3] Shanghai Tianrang Intelligent Technology Co., Ltd
kai.xu@tianrang-inc.com

**Abstract.** Traffic signal control in cities today is not well optimized according to the feedback received from the real world. And such an inefficiency in traffic signal control results in people's waste of time in commuting, road rage in the traffic jam, and high cost for city operation. Recently, deep reinforcement learning (DRL) approaches shed lights to better optimize traffic signal plans according to the feedback received from the environment. Most of these methods are evaluated in a simulated environment, but can not be applied to intersections in the real world directly, as the training of DRL relies on a great amount of samples and takes a long time to converge. In this paper, we propose a batch learning framework where the targeted transfer reinforcement learning (*TTRL-B*) is introduced to speed up learning. Specifically, a separate unsupervised method is designed to measure the similarities of traffic conditions to select the suitable source intersection for transfer. The proposed framework allows batch learning and this is the first work to consider the impact of slow learning in RL on real-world applications. Experiments on real traffic data demonstrate that our model accelerates learning with good performance.

**Keywords:** Deep Reinforcement Learning · Transfer Learning · Traffic Signal Control.

## 1 Introduction

Traffic congestion is one of the most severe issues in cities today. Part of the reason is that the current traffic signal system is not efficient. Current traffic signal control systems such as SCATS [9] and SCOOT [7] adjust traffic signals locally according the loop sensor data at the intersection and they do not optimize globally based on the feedback received from the real world. Recent attempts using Deep Reinforcement Learning (DRL) have shown more effective results [25,24,4,13]. Compared with traditional transportation approaches,

* Work done during an internship at Tianrang

DRL approaches can learn and adjust traffic signal policy based on the feedback received from the environment.

However, if we directly apply DRL to traffic signal control problem, we face two key challenges: (1) the training of a DRL model usually requires millions of samples [12], but we usually have very limited data on a new real-world intersection; (2) the principle of RL is trial-and-error and such error may cause severe implications in the real world. Therefore, we ask a critical question: how can we transfer the knowledge learnt from other intersections to this new intersection so we can try to reduce the error and speed up the learning process?

Transfer learning [14,19] and meta-learning [21,22] have been widely used to transfer knowledge from similar tasks to speed up the learning of target tasks. Recently, researchers apply this idea in DRL to play games [18,15,20]. In these problems, agents learning from different games separately will act as teachers to distill knowledge in various ways, e.g., policy regression [18,15] and high-level feature representation regression [15]. A student model may take over these knowledge and adapt itself while interacting with the new environment. However, such a useful approach has never been investigated in traffic signal control scenario.

In this paper, we propose a transfer learning model for traffic signal control on a series of intersections. Our model is an organic combination of three steps: (1) source task selection; (2) model and sample transfer; (3) a batch learning framework.

*We first select proper source tasks for target using the similarity of embeddings of traffic volume variation.* This can effectively avoid the negative transfer [15]. Previous methods either use domain knowledge [3] or rely on a joint learning model with a task classifier and a RL agent [20]. In our problem setting, using traffic data to measure the similarity is more accurate than using domain knowledge. We also find training a joint model requires much more data samples and it is significantly slow. *Second, we transfer the model and samples to the target intersection*: besides employing the model which mimics the teachers' actions as the pretrained model for the new intersection, we further refer to the teachers' samples to regulate the parameter update when applied to new intersections. *Third, we adopt a batch learning framework to further improve the knowledge distillation.* In each round, well-tuned transfer models are saved in a teacher pool. In the next round, these transferred models will also play the role of teachers. This will keep on distillating the knowledge to its most concise representation.

Our contributions can be summarized as follows:

- This is the first work to consider the effective transfer of RL algorithms trained on simulated traffic to the real-world traffic. This is essential to reduce the mistakes to be made in the real world.
- We propose an elegant transfer learning framework with unsupervised teacher selection and batch learning.

– We conduct comprehensive experiments on the real-world traffic datasets from Hangzhou, China. We show that our proposed method outperforms the baselines and each component of the proposed method makes its own contribution.

## 2 Related Work

### 2.1 Approaches for Traffic Signal Control

**Traditional Transportation Approaches** The current road traffic is mainly managed by systems with two kinds of control: fixed-time [11,23] or vehicle-actuated signals [23,2]. Fixed-time control gives a fixed cycle and green ratio split, while vehicle-actuated determines the time to change signals according to a specific rule (e.g., whether the number of vehicles on the red direction is larger than a threshold). Some other transportation practice [17] also suggests to use the historical traffic volume to compute the cycle and green ratio split, in order to minimize the total travel time under certain traffic volume assumptions. However, those methods all depend heavily on either manually crafted rules or unrealistic assumptions. The policy that achieves good performance on one intersection cannot be applied to another efficiently, either.

**Reinforcement Learning Approaches** RL approaches have been proved to achieve better performance in traffic signal control in recent studies. Early studies [25,1] used tabular methods to compute the reward for discrete state-action pairs. Unfortunately, continuous traffic attributes or high-dimensional features were never fully exploited. Recent deep reinforcement learning methods [24,8,4,13] further utilize the continuous traffic features to solve the problem. However, all these methods treat intersections as individuals, in which model parameters are learned from scratch. As a result, experience accumulated on previous intersections can not be utilized to speed up the learning on new intersections. This will result in slow learning and economic loss in real practice.

### 2.2 Methods for Knowledge Transfer

Transfer learning [14,19] and meta learning [21,22] are methodologies that people proposed to share the knowledge among tasks to boost the performance or speed up the learning. With transfer algorithms as key components in both methods, meta learning concentrates more on a continual stream of tasks while transfer learning may reasonably focus on a single pair of related tasks. Recently, they have been proved to benefit RL learning practices in many game tasks, e.g., Atari [18,15], Minecraft [20], etc. However, little efforts have been made to transfer the learning of traffic signal control problems to mitigate the real traffic congestion problem. Compared to the other transfer learning problems, learning to control traffic signals is cost-sensitive so that the transfer source and target need to be more carefully selected and the transferred knowledge needs to be better represented to avoid negative transfer [3,20,15]. Therefore, we need to develop a new transfer learning framework in this paper.

## 3 Problem Definition

In a single intersection with four-way traffic, there is a signal to direct the traffic. There are two kinds of traffic light settings and we call them phases, i.e., *Green-Horizon* (green light on the horizontal direction and red light on the vertical direction), *Red-Horizon* (red light on the horizontal direction and green light on the vertical direction).

Projecting the situation to the RL definitions, the traffic condition on this intersection, such as the position and speed of each vehicle, is treated as the environment. An agent is trained to decide whether to change the signal to the next phase (action is 1) or keep the current phase (action is 0). In each time slot, the agent takes an action, and receives a reward from the environment. Then, the agent updates the model after a certain period.

*Problem 1.* The goals of this paper are:

- Design a RL algorithm to control the traffic signal to minimize the total travel time of vehicles.
- Transfer the knowledge accumulated in learned intersections to the target intersections to speed up agent learning.

## 4 Method

Our model is a transfer learning solution to speed up learning in target tasks with experience accumulated in source tasks. In this section, we will first introduce a non-transfer RL method *IntelliLight* for signal control. Then we show the transfer properties of our model in three aspects: (1) source task selection; (2) model and sample transfer; (3) the batch learning transfer framework.

### 4.1 Non-transfer Reinforcement Learning Solution

Our signal control model *TTRL-B* follows the agent design and the network structure of model *IntelliLight* [24]. This non-transfer model is a DQN [12] solution and has two additional techniques, i.e., Memory Palace and Phase Gate, to enhance model performance. The agent takes the action with the maximum long-term reward and updates at the $i$-th iteration according to the following loss function:

$$L(\theta_i) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim U(\mathcal{D})}\left[\left(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_i^-) - Q(s_t, a_t; \theta_i)\right)^2\right], \quad (1)$$

in which $\gamma$ is the discount factor, $\theta_i$, $\theta_i^-$ are the parameters of the Q-network at $i$-th iteration for action prediction and for target computation, respectively, $\mathcal{D}$ is the pool of stored samples.

### 4.2 *TTRL-B*: Targeted Transfer Reinforcement Learning in a Batch Learning Framework

To control the signal for a target traffic flow, the proposed model first looks for the most similar flows by analyzing their distance from the target in the embedding space. Then the model for the target task is built with weights initialized via model guidance and keeps on updating with sample guidance. To control signals on a set of intersections, *TTRL-B* will create batches of target tasks to form a batch learning framework.

**Source Task Selection Based on Traffic Embedding** One policy, that successfully eases a congested intersection, plays a instruction role for controlling another high-traffic intersection. Given historical traffic condition of an intersection without deterministic labels, we treat targeted source selection as an unsupervised task where traffic similarities are measured by their distance from each other in an embedding space. Traffic flows are time series data with the number of passing vehicles over a certain time interval in each direction periodically recorded. To represent flow data of an arbitrary length by a fixed-dimensional vector, we build a long short-term memory (LSTM) [6] autoencoder to produce a dense representation that captures the road volume variation along time. In particular, the autoencoder consists of one encoder that first maps the sequence input to a fixed-dimensional vector, followed by one decoder that inversely reconstructs the original sequence. The reconstruction loss between the original and the generated sequence is minimized and we finally extract the state vector of the encoder at the final time step as the traffic representation. Note that the flow information on the intersection to be controlled is unknown, we replace it with the historical traffic data on this intersection from the same time period in the identical workday (or weekday) to represent the upcoming traffic condition.

As the euclidean distance among vectors is widely adopted for their similarity calculation [10,26], we calculate such distance between the target flow representation with that of the candidate source flows, each of which is controlled by an agent with a rich accumulation of samples and experience. $k$ among the source candidates, which are closest to the target in the embedding space, are selected and their respective agents will transfer knowledge to the target agent.

**Transfer Reinforcement Learning**

**Model Guidance.** Given a set of source flows $F_1, \ldots, F_k$, the first step is to train a single network that can control signals of the source flows under the supervision of a set of DQN agents $A_1, \ldots, A_k$, which were once responsible for the source tasks. Agent $A_i$ has a sample pool $\mathcal{D}_i^S = \{(s_t, Q^S(s_t, a)\}$, where the sample from the $t$-th time step consists of the current state $s_t$, and a vector $Q(s_t, a)$ of unnormalized Q-values with one value per action. The target network is trained with a mean-squared-error loss (MSE) that would match Q-values between the

source and target network:

$$L_{MSE}(\theta) = \sum_{i=1}^{i=k} \sum_{(s_t, Q^S(s_t,a)) \in \mathcal{D}_i^S} \left\| Q^S(s_t, a) - Q^T(s_t, a) \right\|_2^2, \tag{2}$$

where $Q^S(s_t, a)$ is sampled from $\{\mathcal{D}_i^S | 1 \leq i \leq k\}$ to represent the Q-value predicted by the source network, $Q^T(s_t, a)$ is the Q-value predicted by the target network parameterized by $\theta$.

For knowledge transfer from the source tasks to the target task, it is possible to replace MSE with other frequently adopted loss functions, e.g., negative log likelihood loss (NLL) [18], cross-entropy loss [15], Kullback-Leibler divergence (KL) [18,5], etc.

As the traffic signal control tasks have the identical state and action space, we directly use the weights of the previously trained target network as an instantiation for a new DQN model that will be trained on the target task. We call such supervised training of the target network as model guidance. Since the source and target flows are very close to each other in traffic embedding, model guidance from source agents will be effective in signal control on the target intersection.

**Sample Guidance.** Previous Experiments show that knowledge transfer from source tasks via model initialization does not always have significant positive effects on the target task [15]. Meanwhile, it has been pointed out when DQN algorithm was first proposed, that deep reinforcement learning tends to be unstable or even diverge for several causes: one is the correlations in the sequence of observations, another is the fact that small updates to $Q$ may significantly change the policy and the data distribution [12]. Applying model guidance alone is likely to cause the same instability problem to DRL in the very beginning of the training, where samples accumulated from the new task are consecutive, limited and biased. One of the approaches for DQN to removing correlations in the observation sequence is to randomize over the data through experience replay. However, sample accumulation for replay memory needs plenty of time followed with great cost in signal control domain. Hence realizing experience replay based on samples from the target task does not benefit the model learning at the very beginning.

We introduce another transfer method called sample guidance, where the replay memory is filled with sufficient samples collected from the source agents' learning process prior to training on target tasks. Through sample guidance, the participation of source networks on the target network is not limited to the parameter initialization, but extended to every subsequent update. Based on the basic DQN update listed in Eq. 1, we define the parameter update for *TTRL-B* at the $i$-th iteration with sample guidance as follows.

$$\begin{aligned} L(\theta_i) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim U(\mathcal{D}^T, \mathcal{D}^S)} \\ \left[ \left( r + \gamma \max_{a_{t+1}} Q^T(s_{t+1}, a_{t+1}; \theta_i^-) - Q^T(s_t, a_t; \theta_i) \right)^2 \right], \end{aligned} \tag{3}$$

where samples are drawn uniformly at random from both the source and target sample pools, i.e., $\mathcal{D}^S = \{\mathcal{D}_i^S | 1 \leq i \leq k\}$ and $\mathcal{D}^T$, respectively.

**A Batch Learning Framework** For a city with all the signals on roads controlled by traditional transportation systems, there is no experience in signal control by RL agents for transfer learning. To resolve such a cold-start problem, we accumulate experience in mediating synthetic flows for fast adaption of RL models to real-world traffic flows.

We believe that knowledge transfer from synthetic to real-world data is better than non-transfer but not the optimal. Synthetic data can hardly mimic every transportation characteristics, while two real flows can have a lot in common, e.g., similar volume trend in daytime, north-east arterial roads, etc. Knowledge transfer between the most similar real-world flows should always be advocated and realized.

Instead of transferring experience of signal control in synthetic flows to all of the real-world intersections, the target intersections are batch selected so that the current batch of roads has an unprecedented amount of source flow candidates than those in the previous batches. In particular, we utilize the Gaussian Mixture Model (GMM) [16] to group all the target flows in $C$ clusters according to their traffic data embedding. Every time we pick the centroid traffic flow in each cluster as one of the target task in the current batch. After determining the source tasks for each target task, *TTRL-B* extracts samples from the source sample pools to learn a DQN model in a supervised way. Learning as well as evaluating on the target traffic flows is conducted with the initial model guidance and the sample guidance in each network update. After the end of each batch, the number of source flow candidates as well as their accumulated experience expands for the next batch of target flows.

## 5 Experiments

We conduct experiments on a simulation platform SUMO (Simulation of Urban MObility)[1]. All the compared algorithms are employed to control the traffic signal on isolated four-way intersections.

### 5.1 Datasets

**Synthetic data.** Vehicles arrive at the approach at uniform rate in the four directions. We utilize 13 different arrival rates which range from 25 to 550 vehicles/hour/lane.

**Real-world data.** We collect the traffic volume data from loop sensors during 04/01/2018-04/30/2018, in Hangzhou, China. There are 48 intersections in total, 22 of which have most sensor undamaged. As the number of vehicles passing

[1] `http://sumo.dlr.de/index.html`

Table 1: Performance evaluated by 2 transfer measures: 1st hour and overall average travel time (in seconds).

| Model | Off-peak Hours | | Peak Hours | |
|---|---|---|---|---|
| | 1st hour | Overall | 1st hour | Overall |
| ***IntelliLight*** | 70.52 | 52.92 | 49.59 | 75.35 |
| ***TTRL-B*** | **35.68** | **32.14** | **34.31** | **70.72** |

one intersection varies dramatically throughout day, evaluation from each passenger's standpoint over a 24-hour time span is not fair for models with good performance on low-density traffic. Therefore, we extract two 5-hour segments from the whole-day traffic, i.e., Off-peak Hours and Peak Hours, and treat them as two separate datasets for a comprehensive model evaluation. Specifically, Off-peak Hours contains continuous traffic flows during which the maximum hourly volume is smaller than 350, while Peak Hours covers those above 350. The average hourly traffic per lane for Off-peak Hours and Peak Hours hours after division is 110.5 and 393.4 respectively.

### 5.2 Compared Methods

We compare the following models to illustrate the benefits of the proposed batch learning framework for targeted transfer. All hyperparameters of the baselines are carefully tuned.

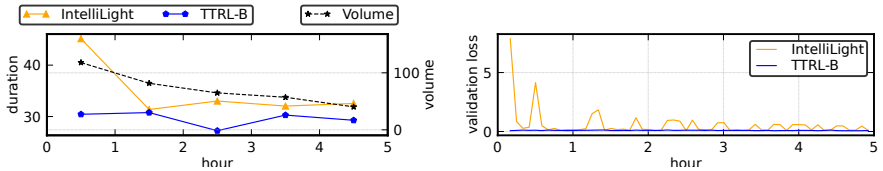
- *IntelliLight* [24]: a recent solution for signaling on the basis of DQN, but with a phase-gated structure to enhance performance.
- *TTRL-B*: our **b**atchwise **t**argeted **t**ransfer **r**einforcement **l**earning based on *IntelliLight*, it maintains an expanding source pool where experience of controlling both synthetic and real-world traffic is accumulated in each batch.

### 5.3 Evaluation Metric

**Average travel time (duration).** Travel time for a vehicle is defined as the time that one car spends from entering the approaching lane until leaving the intersection. We use the average travel time to evaluate different methods.
**Transfer Evaluation.** To measure the effects of transfer, we follow the two metrics suggested in [19]: jumpstart and transfer reward. Under this scenario, these measurements correspond to 1st hour performance and overall performance.

### 5.4 Overall Performance

The results on real-world data are shown in Table 1. As expected, *IntelliLight* with randomly initialized parameters results in long travel time in the 1st hour




(a) Hourly volume of the intersection and duration of the passing vehicles.

(b) Validation loss during training non-transfer and transfer learners.

Fig. 1: Case study of non-transfer and transfer models in Off-peak Hours on Moganshan Road and Wenyi Road in Hangzhou on April 2nd, 2018. We use this sampled intersection throughout this paper for case study.

and the performance gradually improves after 5-hour training. In contrast, *TTRL-B* shows quick adaptivity, with a lower travel time obtained in the 1st hour and in the whole testing process as well. To better demonstrate the fast convergence and adaptability of the proposed model, we show the travel time of vehicles and the validation loss along time for the non-transfer and transfer RL models in Fig. 1a. Compared to the non-transfer model *IntelliLight*, *TTRL-B* always mediates the traffic better from the very start to the end with extremely low loss.

### 5.5 Variants of our model

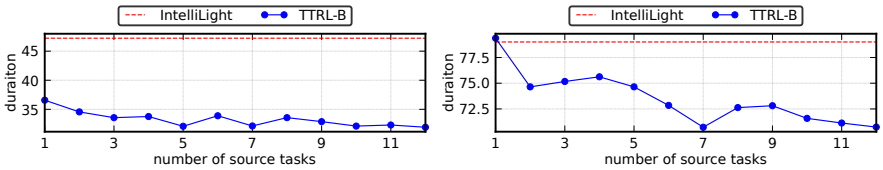
To test effectiveness of the components in our model, we conduct experiments with the following variational models of *TTRL-B*:

- *RTRL-B*: a non-targeted transfer learner, which selects the source tasks randomly regardless of their similarity with the target.
- *TTRL-*{*sample*}: a targeted transfer learner in which sample guidance is removed deliberately.
- *TTRL-*{*model*}: a targeted transfer learner that lacks model guidance in knowledge transfer.
- *TTRL*: a targeted transfer learner without the batch learning framework, so that each target task only has a fixed number of source candidates whose experience is limited in synthetic traffic.

As shown in Table 2, none of the four variants can achieve comparable performance as *TTRL-B*. *RTRL-B* shows inferior performance as the experience from random source flows is not necessarily beneficial to the target. Model guidance alone (*TTRL-*{*sample*}) works fine only in Off-peak Hours (compared to *IntelliLight* in Table 1), while *TTRL-*{*sample*} gets trapped in serious negative transfer in both off-peak and peak hours. It has been proved that sample guidance and model guidance should be combined. Without the batch learning structure, *TTRL* also shows inferior results than *TTRL-B*, due to the never-expanded, experience-limited source pool.

Table 2: Overall performance of four variants of *TTRL-B*.

| Model | Off-peak Hours | Peak Hours |
|---|---|---|
| ***RTRL-B*** | 39.47 | 77.82 |
| ***TTRL-{sample}*** | 34.50 | 76.64 |
| ***TTRL-{model}*** | 50.04 | 82.63 |
| ***TTRL*** | 33.27 | 73.42 |
| ***TTRL-B*** | **32.14** | **70.72** |




(a) Performance in Off-peak Hours. (b) Performance in Peak Hours.

Fig. 2: Parameter sensitivity of model *TTRL-B* in the number of source tasks.

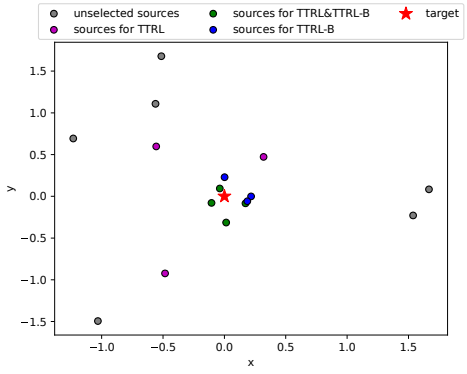
### 5.6 Parameter sensitivity

As shown in Figure 2, our method achieves the best performance when experience from 7 source tasks are used to train the model. But generally, it is not sensitive to the number of source tasks as the travel time on intersections controlled by *TTRL-B* is always far below that of the non-transfer model *IntelliLight*.

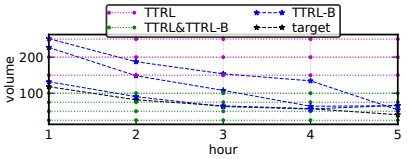
### 5.7 Case study of the Batch Learning Framework

To show the efficiency of knowledge transfer in the batchwise way in detail, we compare *TTRL-B* with the plain targeted transfer learner *TTRL*, which only has experience guidance from synthetic flows. In Fig. 3, we show the comparison of *TTRL-B* and*TTRL* in three aspects: traffic embedding, volume trends and periodical performance.

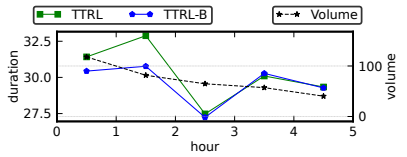
To visualize the relationships between flows, we map the high-dimensional traffic embedding in a 2-dimensional space. Figure 3a shows that both of two targeted transfer learners select source tasks that deal with traffic flows in a relative small euclidean distance to the target's flow. *TTRL-B* differs from *TTRL* as the former retains the most similar synthetic sources selected by *TTRL* and adds some close real-world ones. Based on the volume trend of source flows along time in Fig. 3b, real-world sources selected by *TTRL-B* seem more reasonable than synthetic ones, as they have many transportation characteristics in common, which can be captured by embeddings, e.g., the tendency of traffic load, the number of vehicles in the same time interval, etc. In Fig 3c, selecting source tasks in the batchwise way further proves effective when controlling signals according to their guidance: *TTRL-B* shows an obvious advantage over *TTRL* in the jumpstart and overall performance in our sampled intersection.




(a) Visualization of traffic embeddings. The target is at the origin while each source is located at a random position of a circle with the target location as center and the distance from the target in the embedding space as radius.




(b) Volume trends of selected flows.




(c) Hourly volume of the intersection and duration of the passing vehicles with signals controlled by different models.

Fig. 3: Case study to analyze benefits of the batch learning framework on one real-world intersection.

## 6 Conclusion

In this paper, we solve the problem of using RL to do the traffic signal control on new intersections. Compared with traditional methods, we propose a batchwise targeted transfer framework, which can significantly speed up the convergence and achieve lower vehicles' travel time with much fewer training samples from the new intersection. This will avoid the high cost of traffic jam when directly applying RL algorithms in real world intersections. Our extensive experiments have shown that our method outperforms the baselines and each component makes contribution to the performance boost. We are going to extend our work to more real scenarios by considering multi-phase (e.g., turning vehicles) and multi-intersection traffic signal control for the future work.

## References

1. Bakker, B., Whiteson, S., Kester, L., Groen, F.C.: Traffic light control by multiagent reinforcement learning systems. In: Interactive Collaborative Information Systems, pp. 475–510. Springer (2010)
2. Cools, S.B., Gershenson, C., DHooghe, B.: Self-organizing traffic lights: A realistic simulation. In: Advances in applied self-organizing systems, pp. 45–55. Springer (2013)
3. Du, Y., Gabriel, V., Irwin, J., Taylor, M.E.: Initial progress in transfer for deep reinforcement learning algorithms. In: Proceedings of Deep Reinforcement Learning: Frontiers and Challenges Workshop, New York City, NY, USA (2016)
4. Gao, J., Shen, Y., Liu, J., Ito, M., Shiratori, N.: Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. arXiv preprint arXiv:1705.02755 (2017)

5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
7. Hunt, P., Robertson, D., Bretherton, R., Winton, R.: Scoot-a traffic responsive method of coordinating signals. Tech. rep. (1981)
8. Liu, M., Deng, J., Xu, M., Zhang, X., Wang, W.: Cooperative deep reinforcement learning for traffic signal control (2017)
9. Lowrie, P.: Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic (1990)
10. Lu, W., Hou, J., Yan, Y., Zhang, M., Du, X., Moscibroda, T.: Msql: efficient similarity search in metric spaces using sql. The VLDB JournalThe International Journal on Very Large Data Bases **26**(6), 829–854 (2017)
11. Miller, A.J.: Settings for fixed-cycle traffic signals. Journal of the Operational Research Society **14**(4), 373–386 (1963)
12. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
13. Mousavi, S.S., Schukat, M., Howley, E.: Traffic light control using deep policy-gradient and value-function-based reinforcement learning. Intelligent Transport Systems (ITS) **11**(7), 417–423 (2017)
14. Pan, S.J., Yang, Q., et al.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2010)
15. Parisotto, E., Ba, J.L., Salakhutdinov, R.: Actor-mimic: Deep multitask and transfer reinforcement learning. arXiv preprint arXiv:1511.06342 (2015)
16. Reynolds, D.: Gaussian mixture models. Encyclopedia of biometrics pp. 827–832 (2015)
17. Roess, R.P., Prassas, E.S., McShane, W.R.: Traffic engineering. Pearson/Prentice Hall (2004)
18. Rusu, A.A., Colmenarejo, S.G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., Hadsell, R.: Policy distillation. arXiv preprint arXiv:1511.06295 (2015)
19. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research **10**(Jul), 1633–1685 (2009)
20. Tessler, C., Givony, S., Zahavy, T., Mankowitz, D.J., Mannor, S.: A deep hierarchical approach to lifelong learning in minecraft. In: AAAI. vol. 3, p. 6 (2017)
21. Thrun, S., Pratt, L.: Learning to learn. Springer Science & Business Media (2012)
22. Wang, J.X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J.Z., Munos, R., Blundell, C., Kumaran, D., Botvinick, M.: Learning to reinforcement learn. arXiv preprint arXiv:1611.05763 (2016)
23. Webster, F.V.: Traffic signal settings. Tech. rep. (1958)
24. Wei, H., Zheng, G., Yao, H., Li, Z.: Intellilight: A reinforcement learning approach for intelligent traffic light control. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). pp. 2496–2505 (2018)
25. Wiering, M.: Multi-agent reinforcement learning for traffic light control. In: Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000). pp. 1151–1158 (2000)
26. Zhang, Z., Huang, K., Tan, T.: Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 3, pp. 1135–1138. IEEE (2006)