Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling (ICAPS 2026)

Planning Under Observation Mismatch for Traffic Signal Control via Adaptive Modular World Models

Zherui Huang, Yicheng Liu, Chumeng Liang, Guanjie Zheng*

Shanghai Jiao Tong University

{huangzherui, liuyicheng1515}@sjtu.edu.cn, caradryan2022@gmail.com, gjzheng@sjtu.edu.cn

Abstract

Deploying learned decision-making systems often requires transferring to new sites where the sensing pipeline differs. In such cases, observations can change in semantics and dimensionality even when action primitives and objectives remain comparable. In this work, we study transferable model-based planning under this observation mismatch, which remains challenging for existing learning-based approaches. We propose Adaptive Modularized Model (AMM), a modular planning architecture that separates a domain-specific observation adapter from a shared internal dynamics model defined in a common planning state space. The dynamics model is meta-learned from multiple source domains to enable fast adaptation with limited target interaction. At run time, AMM performs receding-horizon planning by rolling out candidate action sequences under the learned dynamics and selecting actions that optimize a task-specific objective over predicted futures. We instantiate the approach on cross-domain traffic signal control, where actions correspond to signal phases and the planning objective captures congestion. Experiments show that AMM improves both performance and data efficiency compared with existing conventional controllers and learning-based baselines.

Code — github.com/ZheruiHuang/AMMforTSC

Datasets — traffic-signal-control.github.io/index.html

Introduction

Automated planning and scheduling systems are increasingly deployed in settings where the control problem repeats across many sites (Ghallab, Nau, and Traverso 2016). A recurring obstacle is that the sensing and logging pipeline changes from site to site. Sensors differ in modality and coverage, while feature definitions differ due to environmental factors (e.g., vendors and jurisdictions) (Xing et al. 2021). As a result, the observation space can change in semantics and dimension even when the available actions and the underlying physical process remain similar (Sun et al. 2022). This form of observation mismatch makes it difficult to reuse learned decision-making components (Zhao, Queralta, and Westerlund 2020). It also raises the cost of deployment, since collecting extensive online interaction data at each new site can be expensive and risky (Garcıa and Fernández 2015; Levine et al. 2020).

Model-based planning offers a natural solution to this challenge (Mayne et al. 2000; Chua et al. 2018). If one can learn a compact planning state and a predictive model that captures the system dynamics, then the controller can evaluate candidate action sequences by rollout and select actions with a receding-horizon plan (Rawlings et al. 2017). Learning can supply the required components by fitting a state abstraction that exposes the variables needed for planning and a dynamics model that is accurate in the state space that matters for control (Ha and Schmidhuber 2018; Hafner et al. 2019b). Recent work on world model planning has shown that decomposing a controller into a representation model, a dynamics model, and a value or prediction model can support strong performance in complex domains (Hafner et al. 2019b,a; Schrittwieser et al. 2020). However, most existing approaches are typically studied with a fixed observation interface and have not been the primary focus for transfer across changing observation pipelines.

This paper studies transferable planning under observation mismatch. We consider a family of related domains that share the same action space and planning objective, and whose latent dynamics are similar, but whose observations differ. Our goal is to obtain a planner for a new target domain using limited target interaction, while leveraging data collected from multiple source domains. The key design choice is to separate what must change with the sensors from what should remain stable across domains. We therefore introduce a modular architecture that decouples a domain-specific observation adapter from a shared internal dynamics model that supports lookahead planning.

We present Adaptive Modularized Model (AMM), an adaptive modular world model planner. AMM maps raw observations to a shared planning state using a domain-specific representation module. It then predicts the transition of that planning state under candidate action sequences using a shared dynamics module. A value module scores predicted future states, which enables receding horizon planning by selecting the action sequence with the best predicted outcome. The shared dynamics module is learned with meta learning across source domains so that it can take advantage of large amounts of multi-source heterogeneous data and be

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

adapted to a new target domain with limited data and few updates. Only the observation adapter must be replaced or retrained when the observation interface changes, which enables data-efficient reuse of the planning model across deployments with heterogeneous sensing.

In this work, we instantiate this framework in traffic signal control problems. Traffic signal control can be viewed as a sequential planning problem (Varaiya 2013; Wei et al. 2019c). At each control interval, each intersection selects a signal phase, and the joint decisions across intersections shape network-wide congestion over time. Traffic systems also exhibit the deployment conditions that motivate our setting (Zhang et al. 2019). Different cities and districts often provide different sensing and logging capabilities (Zang et al. 2020). Some provide only coarse counts, while others provide speeds or trajectory aggregates. At the same time, the action primitives and the underlying queueing dynamics have a common structure across deployments (Sun et al. 2024). These properties make cross-domain traffic signal control a suitable testbed for studying learning components that enable planning under observation mismatch (Zhang et al. 2019; Wei et al. 2019c; Jiang et al. 2024).

We evaluate AMM on three public benchmarks with heterogeneous observations and limited target interaction. Across all road networks, AMM improves both average travel time and average queue length compared with conventional controllers and strong learning-based baselines. We further conduct ablations that isolate the effects of modularization and meta-learning, as well as analyses on model capacity, source-domain choice, and a case study that trains the observation adapter from logged data.

This work makes three main contributions. First, we formalize transferable planning under observation mismatch, where observation spaces vary across domains while action semantics and objectives are shared. Second, we propose AMM, a modular world model planner that combines a domain-specific observation adapter with a shared meta-learned dynamics model for planning. Third, we evaluate AMM on traffic signal control problems and provide ablations that isolate the effects of modularization, meta-learning, and data budget on target-domain adaptation.

The remainder of this paper is organized as follows. We first review related work on learned planning models, transfer under observation mismatch, and traffic signal control. We then present AMM, including the modular world model, the meta-learning procedure, and the receding-horizon planner. Next, we report experimental results and analyses. Finally, we conclude with limitations and directions for future work.

Related Work

Planning with learned models A central theme in learning for planning is to acquire predictive models and abstractions from data, and then use them for online decision-making. Model-based reinforcement learning has explored planning by rollout with learned dynamics, including probabilistic ensembles and trajectory optimization (Chua et al. 2018), as well as latent world models that enable imagination-based control (Hafner et al. 2019a,b). MuZero popularized a representation, dynamics, and prediction decomposition and combined it with search (Schrittwieser et al. 2020). Our approach is aligned with this family of methods in its use of a learned latent dynamics model to support lookahead. The main difference is the target setting. We focus on transfer across domains with heterogeneous observations, and we structure the model so that observation handling is domain-specific while the internal dynamics used for planning are shared and adapted with limited target updates.

Transfer and adaptation under observation mismatch Transfer in reinforcement learning is often studied through domain randomization and augmentation (Tobin et al. 2017), as well as representation learning methods that aim to extract task-relevant factors that generalize across domains. Meta learning provides a complementary approach by learning initial parameters that can be adapted quickly to new tasks (Finn, Abbeel, and Levine 2017). Several works address adaptation when the observation interface changes by learning representations that align across domains or by learning latent states that preserve controllable structure. These methods motivate the separation of perception from dynamics and decision making. Our work follows this principle, but it enforces it at the level of planning by learning a shared dynamics model in a common planning state space, while allowing the observation adapter to vary with the sensing pipeline.

Traffic signal control Traffic signal control has a long history in transportation research. Classical methods include fixed time control (Allsop 1971), self-organizing traffic lights (Gershenson 2004), and MaxPressure control (Varaiya 2013). Deep reinforcement learning has been applied extensively to TSC and has produced strong results in simulation. Representative methods include IntelliLight (Wei et al. 2018), FRAP (Zheng et al. 2019), PressLight (Wei et al. 2019a), CoLight (Wei et al. 2019b), MPLight (Chen et al. 2020), AttendLight (Oroojlooy et al. 2020), MetaLight (Zang et al. 2020), and UniLight (Jiang et al. 2022). Recent work has also emphasized improved state representations for pressure-based control (Wu et al. 2021; Zhang et al. 2022). CityFlow provides a widely used open simulator and benchmark suite for large-scale evaluation (Zhang et al. 2019). LibSignal further standardizes datasets, environments, and implementations for more reproducible comparisons (Mei et al. 2024).

Cross-domain and data-efficient TSC The cost of online exploration has motivated research on cross-domain transfer and offline to online learning in TSC. Recent cross-city methods leverage offline data and limited target interaction to improve deployability (Sun et al. 2024; Jiang et al. 2024). Other lines study offline reinforcement learning for TSC (Zhang et al. 2023; Bokade and Jin 2025) and practical constraints such as safety and missing or partial observations (Du et al. 2023; Mei et al. 2023). These works support the importance of transfer and limited data adaptation in realistic deployments. Most existing approaches focus on transferring policies or coordinating representations under a

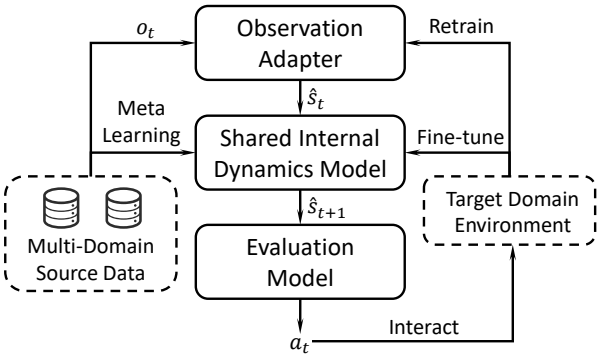

Figure 1: Framework of Adaptive Modularized Model

fixed observation interface. In contrast, our method explicitly targets observation function mismatch by modularizing the observation adapter and transferring the internal dynamics used for lookahead planning.

Method

Problem Formulation

We study transferable model-based planning across a set of related domains. Each domain $d \in \mathcal{D}$ is a controlled process with a shared action space $\mathcal{A}$ and an objective that is comparable across domains. At time t, the agent receives an observation $o_t \in \mathcal{O}_d$ and selects an action $a_t \in \mathcal{A}$. The observation space $\mathcal{O}_d$ may differ in semantics and dimensionality across domains. This captures changes in sensing and preprocessing pipelines.

We assume there exists a shared *planning state space* $\mathcal{S}$ that is sufficient for decision making. The planning state is not directly observed. Instead, each domain provides observations through a domain-specific observation process. Our goal is to obtain a planner for a target domain $d^\star$ using limited target interaction, while leveraging data from a set of source domains $\mathcal{D}_S$.

AMM addresses this setting by learning (i) a domain-specific observation adapter that maps observations into $\mathcal{S}$, and (ii) a shared internal dynamics model in $\mathcal{S}$ that supports online lookahead. At deployment time, AMM performs receding-horizon planning by rolling out candidate action sequences using the learned dynamics and selecting actions that optimize a planning objective over predicted futures.

Adaptive Modular World Model Planning

AMM separates components that depend on the observation interface from components that should transfer across domains. It contains three parts. The first part is an observation adapter. The second part is an internal dynamics model. The third part is an evaluation model used for planning. The framework of AMM is illustrated in Figure 1.

Observation Adapter The observation adapter is domain-specific. It maps a domain observation to a shared planning state representation

$$\hat{s}_t = f_{\phi_d}(o_t), \qquad \hat{s}_t \in \mathcal{S}. \tag{1}$$

The parameters ϕ_d are allowed to vary across domains to accommodate heterogeneous observation spaces. This is the only component that must be replaced or retrained when the observation interface changes.

Shared Internal Dynamics Model The internal dynamics model is shared across domains. It predicts how the planning state transitions under actions as

$$\hat{s}_{t+1} = g_\theta(\hat{s}_t, a_t). \tag{2}$$

For a planning horizon H, we roll out iteratively

$$\hat{s}_{t+h+1} = g_\theta(\hat{s}_{t+h}, a_{t+h}), \quad h = 0, \dots, H-1. \tag{3}$$

The parameters θ are shared across all source domains and the target domain. They are learned from source-domain data and adapted in the target domain with few updates.

Evaluation Model and Receding-Horizon Planning AMM uses an evaluation model to score predicted futures. We denote the evaluation function by $V : \mathcal{S} \to \mathbb{R}$. It can be a learned value model or an explicit objective proxy. Given a candidate action sequence $\mathbf{a}_{t:t+H-1} = (a_t, \dots, a_{t+H-1})$, AMM rolls out (3) and computes

$$\mathrm{Score}(\mathbf{a}_{t:t+H-1}) = \sum_{h=1}^{H} \gamma^{h-1} V(\hat{s}_{t+h}), \tag{4}$$

where $\gamma \in (0, 1]$ discounts future outcomes within the horizon. AMM selects the sequence with the highest score and executes the first action

$$\mathbf{a}^\star_{t:t+H-1} = \arg\max_{\mathbf{a}_{t:t+H-1} \in \mathcal{A}^H} \mathrm{Score}(\mathbf{a}_{t:t+H-1}), \quad a_t \leftarrow \mathbf{a}^\star_t. \tag{5}$$

This produces receding-horizon planning by rollout in a learned planning state space.

Learning and Adaptation

AMM learns the domain adapters $\{f_{\phi_d}\}$ and a transferable initialization of the shared dynamics parameters θ.

Losses We train the dynamics model with multi-step prediction loss in the planning state space. For a trajectory segment starting at time t, let $\hat{s}_{t+h}$ be the rollout produced by (3). We minimize

$$\mathcal{L}^{\mathrm{dyn}}_d(\theta) = \mathbb{E}\Big[\sum_{h=1}^{H} \lambda^{h-1}\, \mathrm{Dist}(\hat{s}_{t+h}, s_{t+h})\Big], \tag{6}$$

where $\lambda \in (0, 1]$ discounts longer rollouts and $\mathrm{Dist}(\cdot, \cdot)$ is a task-dependent distance on planning states.

In our instantiation, the planning state s_t can be computed from the simulator, which allows supervised training of the observation adapter. We minimize

$$\mathcal{L}^{\mathrm{repr}}_d(\phi_d) = \mathbb{E}\Big[\mathrm{Dist}\big(f_{\phi_d}(o_t), s_t\big)\Big]. \tag{7}$$

Algorithm 1: Meta-learning the dynamics initialization

Require: Source domains $\mathcal{D}_S$, step sizes α, β
Initialize dynamics parameters θ
while not converged **do**
  Sample meta-batch $\mathcal{B} \subset \mathcal{D}_S$
  for each domain $d \in \mathcal{B}$ **do**
    Sample support set $\mathcal{S}_d$ and query set $\mathcal{Q}_d$
    $\theta'_d = \theta - \alpha \nabla_\theta \mathcal{L}_d^{\mathrm{dyn}}(\theta; \mathcal{S}_d)$
  end for
  $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{d \in \mathcal{B}} \mathcal{L}_d^{\mathrm{dyn}}(\theta'_d; \mathcal{Q}_d)$
end while
return θ

Algorithm 2: AMM adaptation and control in the target domain

Require: Source domains $\mathcal{D}_S$, target domain $d^\star$
Require: Horizons H, step sizes α, β, η
Meta-learn θ using Algorithm 1
Initialize target adapter parameters $\phi_{d^\star}$
while target interaction budget not exhausted **do**
  Collect target data and update $\phi_{d^\star}$ by minimizing $\mathcal{L}_{d^\star}^{\mathrm{repr}}(\phi_{d^\star})$
  Fine-tune θ by minimizing $\mathcal{L}_{d^\star}^{\mathrm{dyn}}(\theta)$
end while
Deploy with receding-horizon planning using (5)

Meta-Learning the Dynamics Initialization We treat each source domain as a task and meta-learn an initialization for θ so that it adapts quickly to a new domain. We use model-agnostic meta-learning (Finn, Abbeel, and Levine 2017). At each meta-iteration, we sample a meta-batch of domains $\mathcal{B} \subset \mathcal{D}_S$. For each $d \in \mathcal{B}$, we split its data into a support set $\mathcal{S}_d$ and a query set $\mathcal{Q}_d$. We compute task-adapted parameters by a gradient step on the support loss

$$\theta'_d = \theta - \alpha \nabla_\theta \mathcal{L}_d^{\mathrm{dyn}}(\theta; \mathcal{S}_d), \tag{8}$$

then update the meta-parameters by minimizing the query losses after adaptation

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{d \in \mathcal{B}} \mathcal{L}_d^{\mathrm{dyn}}(\theta'_d; \mathcal{Q}_d). \tag{9}$$

The sum in (9) is over domains in the meta-batch. It is not a sum over minibatches within a single domain. The meta-learning procedure is shown in Algorithm 1.

Target-Domain Adaptation and Deployment Given a target domain $d^\star$, we initialize the dynamics model with the meta-learned parameters θ. We then learn the target adapter $f_{\phi_{d^\star}}$ and fine-tune θ using a limited interaction budget in $d^\star$. After adaptation, AMM controls the target domain by applying (1) and (5) at each decision step. The procedure of target-domain adaptation is shown in Algorithm 2.

Instantiation to Traffic Signal Control

We now specify how the formulation and modules above are instantiated for traffic signal control.

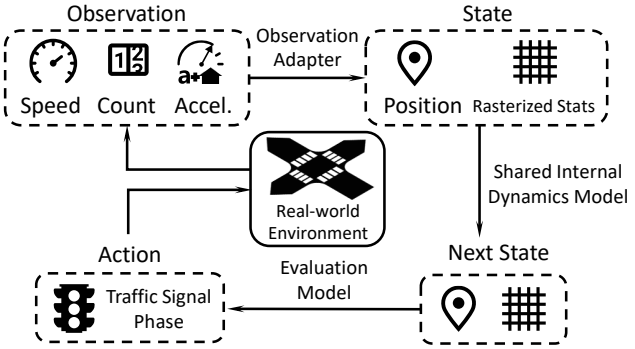

Figure 2: Illustration of instantiation to traffic signal control

Domain and actions A domain corresponds to a road network and traffic flow configuration with a particular observation interface. The road network contains M signalized intersections. At each control interval t, intersection i selects an action a_t^i from a discrete phase set $\mathcal{A}^i$. AMM uses parameter sharing across intersections. It applies the same model parameters to every intersection and outputs one action per intersection at each step. Planning is performed per intersection using (5), and it can be executed in parallel across intersections.

Planning state For each intersection i, the planning state $s_t^i \in S_i$ lies in a fixed domain-invariant space. In this work, we consider two components of s_t^i: (1) normalized vehicle positions near the stop line for each incoming lane, and (2) rasterized approach-level statistics over segments, including estimated vehicle counts and average speeds.

Observation mismatch Each domain d exposes an observation vector $o_t^{i,d}$ for each intersection. The observation components differ across domains in dimension and semantics, such as counts, speeds, or flow aggregates. The adapter f_{ϕ_d} maps these heterogeneous observations to $\hat{s}_t^i$, so that the shared dynamics model g_θ operates in a consistent planning state space across domains.

Evaluation function We score predicted futures using a congestion-oriented value proxy. We aggregate occupancy in n-cell segments and apply larger weight to segments closer to the stop line. This defines $V(s)$ and $\mathrm{Dist}(\cdot, \cdot)$ used in (4), (6), and (7). Details are provided in the experimental section and match the objective of reducing congestion and queueing near intersections.

Experiments

Experimental Setup

Simulator and Benchmarks We use CityFlow (Zhang et al. 2019), an open-source microscopic traffic simulator designed for large-scale traffic signal control research. CityFlow provides efficient simulation of vehicle movements on real road networks and has been widely used in recent TSC benchmarks.

We evaluate on three public benchmarks derived from real traffic data and road networks (Zhang et al. 2019; Wei et al. 2019c). The benchmarks are Hangzhou (4×4), Manhattan (16×3), and Manhattan (28×7), where $x \times y$ denotes the network size, i.e., the number of signalized intersections is roughly arranged in a grid with x rows and y columns. Each benchmark includes a road network file and a vehicle flow file of 3600 seconds. We treat each benchmark as a separate domain. Although two benchmarks are from Manhattan, they correspond to different network sizes and traffic patterns, and they constitute distinct domains in our evaluation.

Observation, State, Value, and Action To study observation-function shift, we assign each domain a different observation vector. All domains include a shared core feature consisting of per-lane vehicle counts near the intersection. We then add a domain-specific feature to induce mismatch. Hangzhou (4×4) provides the number of vehicles that entered the intersection during the last control interval. Manhattan (16×3) provides the number of vehicles passing the midpoint of each road segment during the last control interval. Manhattan (28×7) provides average speeds measured in two road segments, namely the middle third and the last third. AMM uses only these observations as input to its adapter. For baselines, we use the default observation definitions from their implementations, since these methods are not designed for varying observation interfaces across domains.

The internal planning state is domain-invariant and is constructed from simulator ground truth for training and evaluation. It has two complementary parts. First, for each incoming lane, we represent every vehicle by its distance-to-stop-line expressed as a percentage of the lane length. Second, we add a rasterized summary over each approach. For every 10 m segment, we record estimated vehicle count, average speed, and related statistics. The combination captures microscopic proximity and mesoscopic flow and is sufficient to support effective planning in our experiments. These choices are representative rather than mandatory. Other encodings can replace them without changing the method.

We use the average queue length to quantify congestion. It is computed over incoming lanes as the mean number of standing vehicles near the stop line. Lower values indicate better traffic conditions and serve as our planning objective proxy and evaluation metric.

At each control interval, each intersection selects one of eight standard signal phases. The chosen phase is held until the next decision step.

Protocol Each episode simulates 3600 seconds. We use a fixed control interval of 20 seconds. At each decision step, every intersection selects one phase and holds it until the next decision step. This results in 180 decisions per episode. Our controller uses parameter sharing across intersections. The same model parameters are applied to every intersection, and the controller outputs one phase action per intersection at each step.

In the transfer, we follow a leave-one-domain-out protocol. For each target domain, the remaining two domains are used as sources. AMM meta-learns the dynamics initialization on the source domains and then adapts to the target domain using a limited interaction budget. We define the full budget as 60 episodes of target domain interaction. Unless otherwise stated, we report results under the full target interaction budget. Besides, we report results under varying target data budgets, considering budgets in $\{5\%, 10\%, 20\%, \ldots, 100\%\}$. When comparing with learning-based baselines, we restrict their training to the same target interaction budget. This reflects the practical setting where a new deployment provides only limited online interaction. It also avoids conflating target performance with additional target data.

For each learning-based method, we run three seeds and report mean and standard deviation. Conventional baselines are deterministic under fixed traffic flows and thus have zero variance in our setting.

Metrics We report two standard metrics. Average travel time measures the mean time a vehicle spends in the network. Average queue length measures the mean number of waiting vehicles. Lower values indicate better performance.

Baselines

We compare against conventional controllers and learning-based controllers.

Conventional controllers. **Fixed-Time** (Allsop 1971) cycles through phases with pre-defined green durations. **SOTL** (Gershenson 2004) selects phases based on local queue pressure heuristics. **MaxPressure** (Varaiya 2013) greedily selects the phase with maximum pressure and is a strong non-learning baseline.

Learning-based controllers. We include representative deep RL methods for network-level control and multi-agent coordination. **CoLight** (Wei et al. 2019b) uses graph attention to coordinate intersections. **MPLight** (Chen et al. 2020) builds on FRAP (Zheng et al. 2019) and uses pressure-based features. **AttendLight** (Oroojlooy et al. 2020) uses attention to aggregate information and learns a universal controller. **MetaLight** (Zang et al. 2020) applies meta-learning to improve adaptation across scenarios. **E-PressLight** (Wu et al. 2021) and **E-MPLight** (Wu et al. 2021) improve pressure-based representations. **A-MPLight** (Zhang et al. 2022) refines the pressure and demand representation. **UniLight** (Jiang et al. 2022) uses a communication mechanism to improve coordination.

Main Results

Table 1 reports the main results on average travel time and average queue length. AMM attains the best overall performance with low variance across all three benchmarks. A consistent observation is that conventional controllers can outperform several learning-based baselines when the available training data are limited. This is expected since conventional controllers do not require exploration. AMM remains competitive in this regime because it transfers a dynamics model from source domains and only adapts a small number of parameters in the target domain.

Methods	Hangzhou (4×4)		Manhattan (16×3)		Manhattan (28×7)	
	Travel Time	Queue Length	Travel Time	Queue Length	Travel Time	Queue Length
Fixed-Time	416.88 ± 0.00	0.96 ± 0.00	535.60 ± 0.00	0.89 ± 0.00	723.08 ± 0.00	1.17 ± 0.00
SOTL	440.66 ± 0.00	1.21 ± 0.00	862.81 ± 0.00	1.54 ± 0.00	769.67 ± 0.00	1.28 ± 0.00
MaxPressure	303.31 ± 0.00	0.29 ± 0.00	250.14 ± 0.00	0.14 ± 0.00	564.36 ± 0.00	**0.70** ± 0.00
CoLight	689.89 ± 49.20	3.53 ± 0.15	853.52 ± 14.66	1.58 ± 0.05	829.18 ± 8.72	1.24 ± 0.06
MPLight	616.67 ± 204.3	2.63 ± 1.58	775.45 ± 116.6	1.50 ± 0.15	829.73 ± 4.93	1.44 ± 0.09
AttendLight	508.43 ± 158.6	1.97 ± 1.35	712.16 ± 201.2	1.26 ± 0.39	736.47 ± 59.58	1.17 ± 0.05
MetaLight	319.56 ± 13.02	0.53 ± 0.20	289.73 ± 127.8	0.34 ± 0.40	558.45 ± 28.64	0.79 ± 0.09
E-PressLight	708.40 ± 73.76	3.50 ± 0.14	786.79 ± 42.53	1.54 ± 0.08	819.29 ± 13.20	1.28 ± 0.10
E-MPLight	511.19 ± 181.3	1.95 ± 1.62	591.18 ± 171.6	1.00 ± 0.46	764.78 ± 102.3	1.11 ± 0.21
A-MPLight	578.57 ± 138.7	2.71 ± 1.32	758.68 ± 160.8	1.31 ± 0.32	736.50 ± 82.14	1.24 ± 0.27
UniLight	681.77 ± 71.37	1.10 ± 0.13	1303.92 ± 56.34	1.16 ± 0.05	1535.36 ± 16.35	1.24 ± 0.02
AMM w/o Mod.	372.00 ± 69.65	1.06 ± 0.70	197.02 ± 26.60	0.07 ± 0.04	704.33 ± 62.80	1.08 ± 0.13
AMM w/o ML	279.95 ± 3.50	0.15 ± 0.02	171.34 ± 3.14	**0.03** ± 0.01	557.65 ± 23.21	0.75 ± 0.10
AMM	**277.95** ± 3.54	**0.14** ± 0.03	**168.36** ± 2.12	**0.03** ± 0.01	**538.93** ± 6.81	**0.70** ± 0.03

Table 1: Comparison with baselines on benchmarks. Metrics are average travel time (s) and average queue length (vehicles), lower is better. Each method is run three times per benchmark and the table reports mean ± std. Bold marks the best result in each column. Conventional controllers are deterministic under fixed flows and therefore show zero variance.

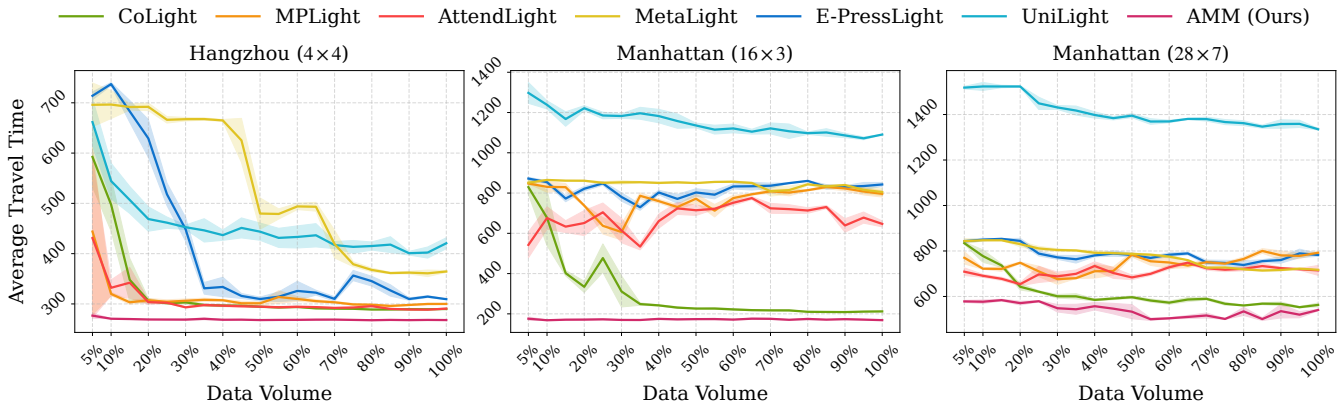

Figure 3: Data efficiency on the target domains. Average travel time as the fraction of the target interaction budget increases, where 100% equals 60 episodes. Curves are averaged over three runs. AMM reaches strong performance with substantially less target interaction than baselines.

Source Domain	Target Domain					
	Hangzhou (4×4)		Manhattan (16×3)		Manhattan (28×7)	
	Travel Time	Queue Length	Travel Time	Queue Length	Travel Time	Queue Length
Hangzhou (4×4)	/		180.67 ± 1.99	0.05 ± 0.01	551.75 ± 13.13	0.72 ± 0.09
Manhattan (16×3)	280.57 ± 2.52	0.16 ± 0.02	/		606.03 ± 62.11	0.76 ± 0.18
Manhattan (28×7)	280.93 ± 2.57	0.16 ± 0.02	174.83 ± 2.02	0.04 ± 0.01	/	
Fixed-Time	416.88 ± 0.00	0.96 ± 0.00	535.60 ± 0.00	0.89 ± 0.00	723.08 ± 0.00	1.17 ± 0.00
MaxPressure	303.31 ± 0.00	0.29 ± 0.00	250.14 ± 0.00	0.14 ± 0.00	564.36 ± 0.00	0.70 ± 0.00

Table 2: Single-source pretraining study. Each row uses the source domain shown to pretrain the dynamics model and then adapts to the target domain with the standard budget. Cells report mean ± std of average travel time and average queue length, lower is better. "/" marks the diagonal where pretraining on the same domain is not applicable. Fixed-Time and MaxPressure are included as reference non-learning controllers.

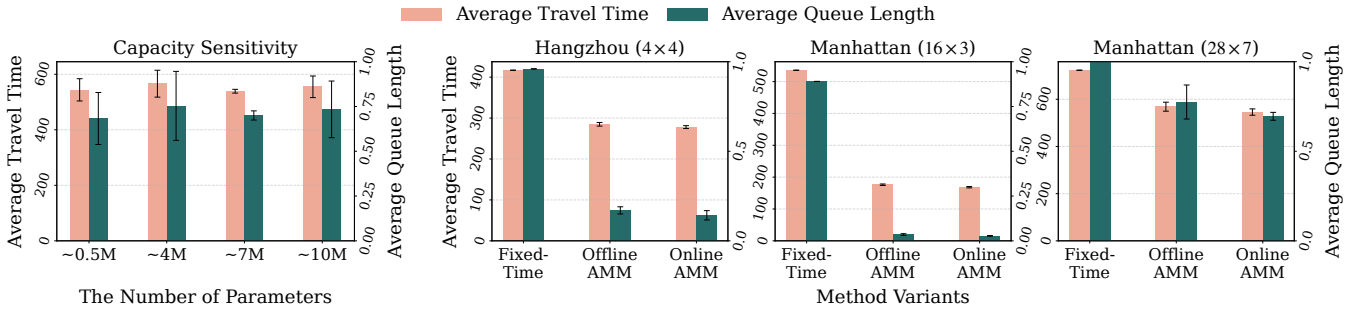

Figure 4: Analysis of AMM. **Left:** Average travel time and average queue length for AMM with different parameter counts. Performance varies little across a wide range of capacities, indicating limited sensitivity to model size under the considered budgets. **Right:** Comparison between offline and online training of the observation adapter. "Offline AMM" fits the adapter from non-interactive logs collected by a fixed-time controller, while "Online AMM" uses online interaction. Offline AMM approaches online performance with only limited logs, highlighting the low-risk adaptation enabled by the modular design.

Data Efficiency Under Limited Target Interaction

Figure 3 compares average travel time as a function of target interaction budget. AMM improves rapidly with small target budgets and reaches strong performance earlier than the baselines. This behavior is consistent with the design of AMM. The dynamics initialization is meta-learned from source domains, and adaptation in the target domain mainly aligns the observation adapter and fine-tunes the dynamics model.

Ablation Studies

We include two ablations to isolate the impact of the modular design and the meta-learning procedure. Results are shown in Table 1.

Effect of modularization The "AMM w/o Mod." represents the non-modularized variant, which uses a single end-to-end model that directly maps observations to value estimates, and it does not separate an observation adapter from the dynamics model. This variant performs substantially worse, particularly under observation mismatch, since it cannot reuse source data in a consistent internal space.

Effect of meta-learning The "AMM w/o ML" represents the sequential training variant that replaces meta learning with sequential multi-domain training. It trains the dynamics model on source domains by standard gradient descent and then fine-tunes on the target domain. Sequential training performs competitively but is consistently worse than AMM. This supports the role of meta-learning in producing an initialization that adapts more effectively with limited target data.

Further Analyses

Sensitivity to model capacity We study model capacity by varying network depth and width and reporting performance as a function of parameter count. Figure 4 left shows that AMM is relatively stable across a wide range of model sizes. Performance saturates once the model reaches moderate capacity, and larger models do not provide consistent additional gains under the considered data budgets.

Source domain selection To assess how source domains influence transfer, we train AMM using a single source domain and then adapt it to the target domain. Table 2 reports the results. Single-source pretraining already yields reasonable performance after target adaptation, which suggests that the learned dynamics capture shared structure. Using both source domains provides the best results, which supports aggregating multi-domain experience.

Training the observation adapter with offline data In many deployments, logged data are easier to obtain than online interaction. We evaluate whether the observation adapter can be trained from logged trajectories collected by a fixed-time controller. Figure 4 right compares Offline AMM, which trains the adapter from logged data, with Online AMM, which trains with interactive target data. Offline AMM attains competitive performance, which indicates that the modular separation enables different training strategies for different components.

Conclusion, Limitations, and Future Work

This paper studies transfer for sequential decision-making when observation interfaces differ across deployments. We cast this setting as transferable model-based planning under observation mismatch and present AMM, a modular world-model planner. AMM separates a domain-specific observation adapter from a shared internal dynamics model in a common planning state space. The dynamics model is meta-learned on source domains for fast target adaptation, and AMM selects actions by receding-horizon planning with learned rollouts.

We instantiate AMM on traffic signal control with heterogeneous observation pipelines. On CityFlow benchmarks, AMM reduces average travel time and queue length compared with conventional controllers and strong learning-based baselines under the same target interaction budgets. Ablations show that modularization enables reuse across observation interfaces and that meta-learning improves adaptation over sequential multi-domain training. Analyses further show stable performance across model

capacities, benefits from aggregating multiple source domains, and that the observation adapter trains effectively from logged data.

Several limitations remain. The planner searches action sequences per intersection with parameter sharing rather than joint actions at network scale. Adapter training uses simulator-derived targets that may be unavailable or noisy in real deployments, and learned rollouts can accumulate error over the planning horizon. Future work will explore joint or factored planning with stronger coupling, uncertainty-aware dynamics and value estimation, weaker or self-supervised adapter learning, and broader evaluations beyond traffic signal control.

Acknowledgments

This work was sponsored by 2025 Shanghai Key Technology Research and Development Program, Next-Generation Information Technology Project under Grant No. 25511103800.

References

Allsop, R. E. 1971. Delay-minimizing settings for fixed-time traffic signals at a single road junction. *IMA Journal of Applied Mathematics*, 8(2): 164–185.

Bokade, R.; and Jin, X. 2025. OffLight: An Offline Multi-Agent Reinforcement Learning Framework for Traffic Signal Control. In *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, 2730–2737. IEEE.

Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; and Li, Z. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3414–3421.

Chua, K.; Calandra, R.; McAllister, R.; and Levine, S. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31.

Du, W.; Ye, J.; Gu, J.; Li, J.; Wei, H.; and Wang, G. 2023. Safelight: A reinforcement learning method toward collision-free traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 14801–14810.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Garcıa, J.; and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1): 1437–1480.

Gershenson, C. 2004. Self-organizing traffic lights. *arXiv preprint nlin/0411066*.

Ghallab, M.; Nau, D.; and Traverso, P. 2016. *Automated planning and acting*. Cambridge University Press.

Ha, D.; and Schmidhuber, J. 2018. World models. *arXiv preprint arXiv:1803.10122*, 2(3).

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2019a. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019b. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565. PMLR.

Jiang, H.; Li, Z.; Wei, H.; Xiong, X.; Ruan, J.; Lu, J.; Mao, H.; and Zhao, R. 2024. X-light: Cross-city traffic signal control using transformer on transformer as meta multi-agent reinforcement learner. In *International Joint Conferences on Artificial Intelligence*.

Jiang, Q.; Qin, M.; Shi, S.; Sun, W.; and Zheng, B. 2022. Multi-agent reinforcement learning for traffic signal control through universal communication method. *arXiv preprint arXiv:2204.12190*.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Mayne, D. Q.; Rawlings, J. B.; Rao, C. V.; and Scokaert, P. O. 2000. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6): 789–814.

Mei, H.; Lei, X.; Da, L.; Shi, B.; and Wei, H. 2024. Libsignal: An open library for traffic signal control. *Machine Learning*, 113(8): 5235–5271.

Mei, H.; Li, J.; Shi, B.; and Wei, H. 2023. Reinforcement learning approaches for traffic signal control under missing data. *arXiv preprint arXiv:2304.10722*.

Oroojlooy, A.; Nazari, M.; Hajinezhad, D.; and Silva, J. 2020. Attendlight: Universal attention-based reinforcement learning model for traffic signal control. *Advances in Neural Information Processing Systems*, 33: 4079–4090.

Rawlings, J. B.; Mayne, D. Q.; Diehl, M.; et al. 2017. Model predictive control: theory, computation, and design, vol. 2. *Madison, WI: Nob Hill Publishing*.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Sun, Q.; Zha, R.; Zhang, L.; Zhou, J.; Mei, Y.; Li, Z.; and Xiong, H. 2024. Crosslight: Offline-to-online reinforcement learning for cross-city traffic signal control. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2765–2774.

Sun, Y.; Zheng, R.; Wang, X.; Cohen, A.; and Huang, F. 2022. Transfer RL across Observation Feature Spaces via Model-Based Regularization. In *International Conference on Learning Representations*.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 23–30. IEEE.

Varaiya, P. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36: 177–195.

Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Gayah, V.; Xu, K.; and Li, Z. 2019a. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1290–1298.

Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; and Li, Z. 2019b. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1913–1922.

Wei, H.; Zheng, G.; Gayah, V.; and Li, Z. 2019c. A Survey on Traffic Signal Control Methods. *arXiv preprint arXiv:1904.08117*.

Wei, H.; Zheng, G.; Yao, H.; and Li, Z. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496–2505.

Wu, Q.; Zhang, L.; Shen, J.; Lü, L.; Du, B.; and Wu, J. 2021. Efficient pressure: Improving efficiency for signalized intersections. *arXiv preprint arXiv:2112.02336*.

Xing, J.; Nagata, T.; Chen, K.; Zou, X.; Neftci, E.; and Krichmar, J. L. 2021. Domain adaptation in reinforcement learning via latent unified state representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10452–10459.

Zang, X.; Yao, H.; Zheng, G.; Xu, N.; Xu, K.; and Li, Z. 2020. Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1153–1160.

Zhang, H.; Feng, S.; Liu, C.; Ding, Y.; Zhu, Y.; Zhou, Z.; Zhang, W.; Yu, Y.; Jin, H.; and Li, Z. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference*, 3620–3624.

Zhang, L.; Wu, Q.; Shen, J.; Lü, L.; Du, B.; and Wu, J. 2022. Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control. In *International Conference on Machine Learning*, 26645–26654. PMLR.

Zhang, L.; Zhang, Y.; Deng, J.; and Li, C. 2023. DataLight: Offline Data-Driven Traffic Signal Control. *arXiv preprint arXiv:2303.10828*.

Zhao, W.; Queralta, J. P.; and Westerlund, T. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, 737–744. IEEE.

Zheng, G.; Xiong, Y.; Zang, X.; Feng, J.; Wei, H.; Zhang, H.; Li, Y.; Xu, K.; and Li, Z. 2019. Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1963–1972.