SPUR – Towards Adaptive Traffic Pattern Clustering Reinforcement Learning in Traffic Signal Control

Online clustering of traffic patterns and cluster-specific RL agents enable safe, scalable and faster-to-deploy traffic signal control under non-stationary demand.

L. Klein, M. Redeker
Mathematical Optimization, Fraunhofer IOSB-INA, Lemgo
{lukas.klein | magnus.redeker}@iosb-ina.fraunhofer.de

Fraunhofer IOSB-INA

SUMO CONFERENCE 2026

Motivation & Challenges

Urban traffic is **highly non-stationary** due to incidents, weather, events, and long-term demand shifts. Fixed-time and actuated signal control struggle to maintain efficiency and safety under such evolving traffic patterns (TPs). Reinforcement-Learning (RL)-based traffic signal control (TSC) is promising [2], but most approaches rely on a **single monolithic controller** that must handle all situations. This limits **learning efficiency, robustness and real-world deployability** in large, heterogeneous networks.

Challenges

- **Non-stationary traffic patterns:** the same average density can arise from incoming vs. receding waves, requiring different control strategies.
- **Monolithic RL policies:** a single model must cover all TPs – hard to interpret and hard to certify.
- **Limited online adaptation:** few mechanisms for continuous, data-efficient on-device learning.
- **Deployment gap:** safety, certification and maintenance effort prevent city-wide rollout [2].

Traffic Pattern Representation

- **Combine core traffic features** such as inflow λ_{in}, outflow λ_{out}, density ρ, and occupancy O.
- **Integrate exogenous factors, e.g.** precipitation, wind speed, temperature, and brightness.
- **De-emphasize time-of-day and day-of-week indicators** to avoid redundancy and increase robustness to incidents.
- **Represent each traffic pattern as a short trajectory T_{env}** over the recent Δ_w minutes using a sliding window.

Fading-Memory Traffic Pattern Similarity

- **Define a distance between recent traffic pattern trajectories** over a fixed history window.
- **Use feature weights** α to emphasize the most relevant signals (e.g., density vs. weather).
- **Apply a fading-memory parameter** k so that recent observations have stronger influence, while older values still capture trend history.
- Consider two traffic patterns similar if their distance is below a threshold ε.

$$d(T_1, T_2; \alpha, k) = \sum_{j=0}^{n} e^{-k(n-j)\Delta r} \sqrt{\sum_i \alpha_i^2 \left(\hat{v}_i^{(1)}[j] - \hat{v}_i^{(2)}[j]\right)^2}.$$

i indexes features, j time steps ($n = \Delta_\omega/\Delta_r$)
α – feature weights, k – fading-memory parameter
$\hat{v}_i^{(1)}[j], \hat{v}_i^{(2)}[j]$ are normalized TP features at time step j

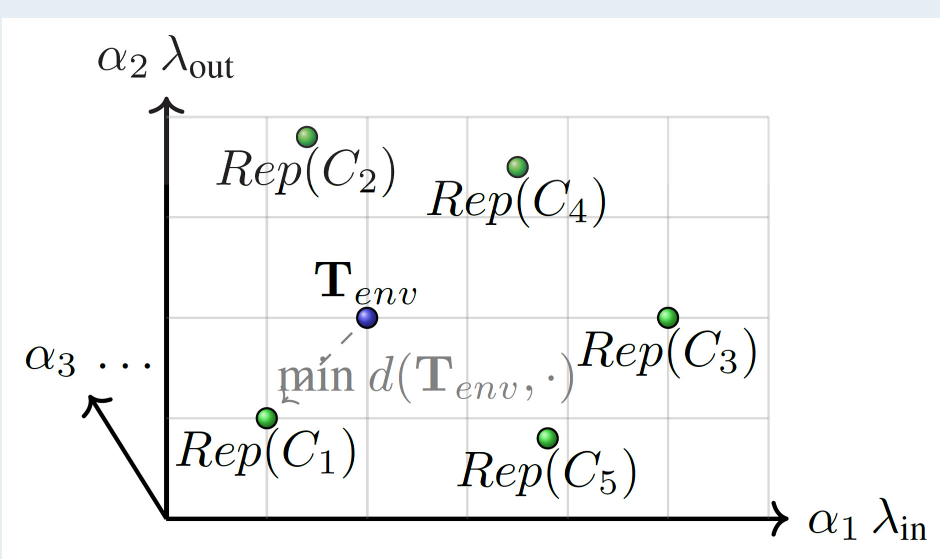

Figure 1: Active traffic pattern clusters and the current environment trajectory T_{env} in the TP feature space.

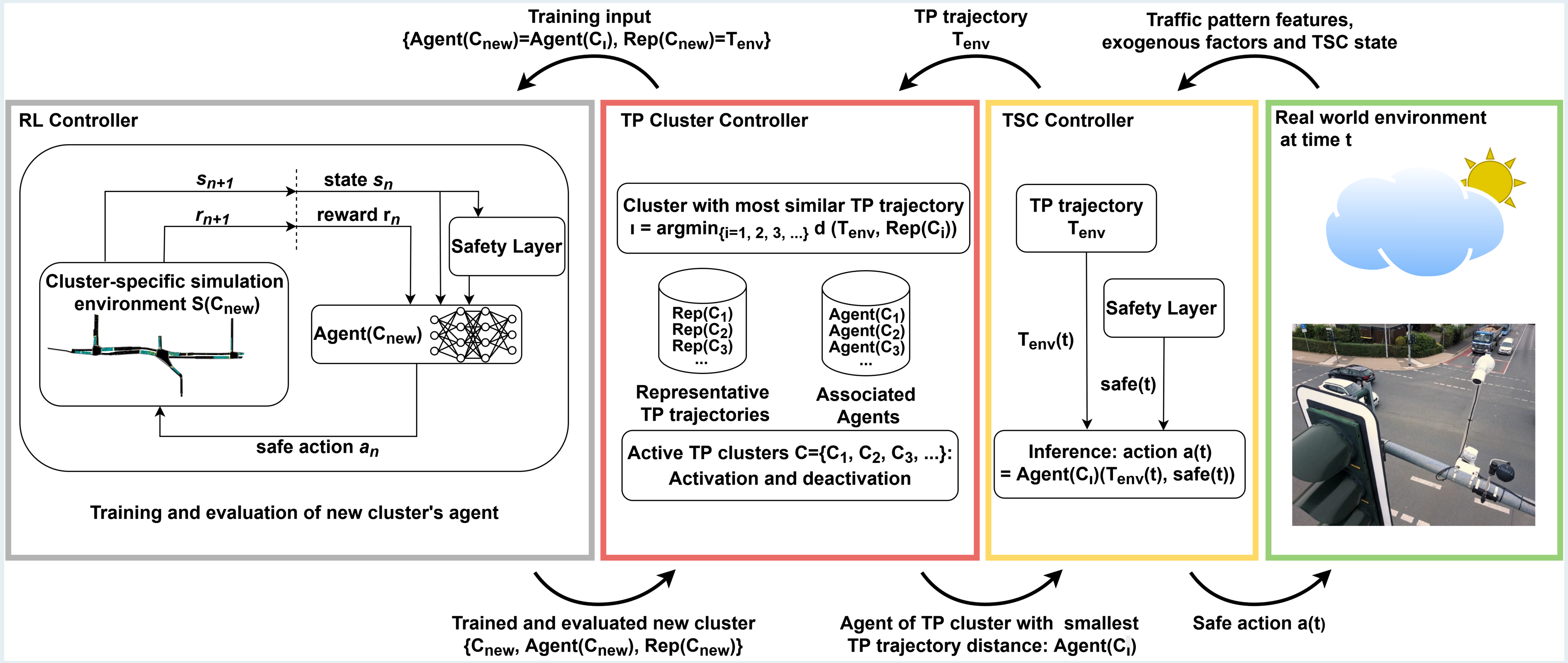

Figure 2: Conceptual overview of the adaptive traffic pattern clustering framework with cluster-specific RL agents for real-time traffic signal control.

Cluster Activation & Transfer-Based Initialization

- **Activate new cluster when no active cluster is similar enough:** If the closest active cluster is still too far $d(T_{env}, Rep(C_ı)) > \varepsilon$ a new cluster C_{new} is activated with $Rep(C_{new}) = T_{env}$.
- **Initialize the new agent by transfer:** Initialize $Agent(C_{new})$ by copying the parameters from the most similar existing $Agent(C_ı)$ (transfer-based initialization).
- **Train the new agent in its cluster-specific simulation:** Train $Agent(C_{new})$ in the cluster-specific simulation $S(C_{new})$ until performance and safety criteria are met.
- **Mitigate negative transfer:** Mitigate negative transfer through conservative learning rates, randomized early training, performance checks, and rollback to a previous policy if necessary.

Cluster Fusion & Budget Control

- **Limit the number of simultaneously active clusters** to $m_{act,max}$.
- When this budget is exceeded, identify the two most similar clusters on the fading-memory distance.
- Compute a **deactivation score** using coverage $Gov(C_i)$ and activation time $Act(C_i)$, and deactivate the less valuable cluster.
- **Keep the more representative cluster** and its agent to maintain a compact, non-redundant portfolio aligned with current traffic conditions.

Adaptive TP-RL-TSC Framework

- **Online TP Clustering**
  - Maintain a small set of active clusters $\{C_1, \dots, C_m\}$, each representing a family of similar traffic patterns.
  - Represent each cluster by a trajectory $Rep(C_i)$ built from recent traffic and exogenous features.
- **Cluster-specific RL agents**
  - Associate each active cluster C_i with its own RL agent $Agent(C_i)$.
  - Train $Agent(C_i)$ in a cluster-specific simulator $S(C_i)$ that is parameterized by $Rep(C_i)$.
- **Online Routing layer**
  - At each control step, route the current environment trajectory T_{env} to its nearest active cluster.
  - Use the corresponding $Agent(C_ı)$ for low-latency signal control decisions in the real world.

Value Proposition

Adaptive TP-RL-TSC **is intended to** deliver concrete benefits for cities, operators, and the RL community.

- **Improved adaptability:** specialized RL agents for distinct traffic patterns instead of a single monolithic controller.
- **Faster time-to-deployment:** new situations can be handled by transfer-initialized agents; clusters can be rolled out incrementally.
- **Reduced computational burden:** the number of active clusters is bounded and pruned; simulators focus on relevant TP families.
- **Operational robustness:** explicit mechanisms to handle extreme events and to switch to rule-based safety controllers when required.

Deployment Pathway

- **Start with a single cluster** representing the most common daily traffic pattern and train its agent Agent(C_1) in simulation.
- **Deploy this agent on one or a few intersections** with continuous monitoring of safety and performance.
- **As new traffic patterns emerge** (e.g., due to weather, events or seasonal shifts), activate new clusters and roll out their cluster-specific agents.
- **Run online clustering and RL training asynchronously** on edge and/or cloud infrastructure, while keeping inference on the controller fast and predictable.
- **Gradually extend the approach** from individual intersections to corridors and city-wide networks.

Conclusion & Outlook

We present a **conceptual modular TP-RL-TSC framework** that combines online traffic pattern clustering with cluster-specific RL agents for real-time traffic signal control. A history-based fading-memory distance is applied consistently across clustering, simulation parametrization and online inference, ensuring coherent treatment of traffic patterns. By maintaining a compact portfolio of specialized agents and refining it online, the approach aims to reduce computational burden and time-to-real-world deployment while improving robustness under non-stationary demand.

Current status: conceptual framework only – implementation and empirical evaluation are planned as future work. A more detailed description of the TP-RL-TSC framework is provided in [1].

[1] L. Klein, M. Redeker. Towards Adaptive Traffic Pattern Clustering Reinforcement Learning in Traffic Signal Control. Preprint, 2026.
[2] A. Müller et al. Towards Real-World Deployment of Reinforcement Learning for Traffic Signal Control. ICMLA, 2022.