

# ST-3CRP: spatio-temporal delay- and fairness-aware cooperative caching with content request prediction in the internet of vehicles\*

Abdelkrim Lamraoui<sup>1,\*†</sup>, Lyamine Guezouli<sup>2,†</sup> and Kamel Barka<sup>3,†</sup>

<sup>1</sup> LAMIE Laboratory, University of Batna 2, Batna, Algeria

<sup>2</sup> LEREESI Laboratory, HNS-RE2SD, Batna, Algeria

<sup>3</sup> Lastic Laboratory, University of Batna 2, Batna, Algeria

## Abstract

Within intelligent transportation systems (ITS), the time-varying and spatially heterogeneous demand of the Internet of Vehicles (IoV) makes cooperative edge caching difficult to optimize using aggregate popularity alone. Existing predictive approaches such as CCCRP combine request prediction with reinforcement learning, but their cache-placement objective is primarily driven by aggregate cache-hit ratio and does not explicitly distinguish heterogeneous delivery paths or service imbalance among vehicle clusters. This paper proposes ST-3CRP, a spatio-temporal, delay- and fairness-aware cooperative caching strategy for IoV. LSTM-based request prediction is conditioned on RSU zones and vehicle clusters to estimate local future demand. A joint cache-placement and service-assignment model distinguishes same-cluster V2V delivery, RSU-based V2I delivery, inter-cluster cooperative delivery, and remote backhaul retrieval. Q-learning evaluates complete feasible cache configurations using a utility that combines expected hit ratio and normalized acquisition delay while accounting for cluster-level service quality. The evaluation uses 200 independent paired runs per operating point and reports 95% confidence intervals. At the most constrained extended cache setting ( $p=0.25$ ), ST-3CRP improves hit ratio over CCCRP by 8.04 percentage points and reduces average delay by 20.03%. Across the coupled cluster-scaling experiment, it improves minimum cluster hit ratio by 4.89 percentage points on average and reduces maximum cluster delay by 13.81% on average.

## Keywords

Internet of Vehicles, intelligent transportation systems, cooperative caching, content request prediction, spatio-temporal caching, LSTM, Q-learning, edge caching, fairness

## 1. Introduction

Connected vehicles increasingly exchange road information, maps, multimedia objects, sensing data, and delay-sensitive services. Serving every request through a remote provider increases backhaul traffic and acquisition delay. Edge computing and multi-access edge computing move storage and computation closer to users [1-4], while wireless caching further reduces repeated core-network transfers by placing popular objects near requesters [5]. In vehicular environments, vehicles with storage can act as caching vehicles (CVs) and roadside units (RSUs) can provide more stable infrastructure caches. Predictive cooperative caching improves this model by using anticipated rather than purely historical demand. The Cooperative Caching Strategy with Content Request Prediction (CCCRP) clusters vehicles according to position and speed, predicts future request counts using LSTM, and applies Q-learning to determine cache placement [6]. Its reported results show advantages over LFU and LRU, but three methodological limitations remain relevant to

\*ADT-ISMDDC 2026: *Advanced Digital Technologies: Intelligent Systems, Modeling, and Data-Driven Control*, September 22–24, 2026, Odesa, Ukraine

<sup>1\*</sup> Corresponding author.

<sup>†</sup> These authors contributed equally.

✉ a.lamraoui@univ-batna2.dz (A. Lamraoui); lyamine.guezouli@hns-re2sd.dz (L. Guezouli); kamel.barka@univ-batna2.dz (K. Barka)

ORCID 0009-0004-2985-8067 (A. Lamraoui); 0000-0002-7406-7633 (L. Guezouli); 0000-0002-6824-1594 (K. Barka)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

heterogeneous IoV scenarios. First, demand may differ substantially across RSU zones and vehicle clusters. Second, a cache hit reached through a local CV, an RSU, or an inter-cluster path is counted identically although the associated delays differ. Third, aggregate hit-ratio maximization can favor high-volume clusters and hide poor service in less dominant clusters.

ST-3CRP addresses these limitations by (i) estimating request popularity jointly in space and time; (ii) separating cache placement from request service assignment so that V2V, V2I, inter-cluster, and backhaul paths are explicitly represented; (iii) optimizing a joint hit-ratio/delay utility; and (iv) introducing a cluster-level service requirement. The Q-learning paradigm is retained, but its reward is aligned with this spatio-temporal, delay-aware, and fairness-aware formulation. The evaluation first reproduces the CCCRP operating scale and then extends it to more heterogeneous settings.

## 2. Related Work

Mobility-aware vehicular caching has been investigated from several complementary perspectives. Yao et al. used mobility prediction to support cooperative content-centric caching [7]. Su et al. jointly exploited RSUs and vehicles and selected content sources according to acquisition delay [8], while Zhang et al. incorporated mobility into vehicular cache optimization [9]. These studies demonstrate that where a content is stored is inseparable from how vehicles move and from which communication path is available.

A second line of work predicts future demand or learns cache-control policies. Echo-state networks have been used for proactive caching with mobile users [10], and deep reinforcement learning has been applied to cache-enabled D2D systems [11]. Qiao et al. formulated cooperative vehicular caching as a double-time-scale learning problem [12], whereas Nan et al. emphasized delay-aware content delivery in IoV [13]. Mobility prediction has also been combined with online bandit learning [14], self-attention and hierarchical federated learning [15], and traffic-aware deep reinforcement learning [16].

More recent work has increasingly coupled mobility awareness, prediction, clustering, and learned cooperative control. Tian et al. developed a mobile-aware cooperative edge-caching scheme using regional popularity estimation and deep reinforcement learning [18]. Chen et al. combined regional traffic features, dynamic vehicle clustering, position prediction, and DRL for cooperative vehicular caching [19]. Zhu et al. integrated content prediction with deep reinforcement learning for vehicular edge-cloud caching [20]. Ghamri et al. proposed a leader-follower multi-agent reinforcement-learning solution for IoV edge caching [21], while Cheng et al. jointly optimized recommendation, caching, and transmission for cooperative vehicular video delivery [22]. ST-3CRP differs from these approaches by combining cluster-conditioned LSTM demand prediction, explicit service-path assignment, and worst-cluster service indicators within the same hit-delay cache-placement objective.

## 3. System Model and Problem Formulation

Consider an urban IoV area served by RSUs  $r_i$ . During cache-update period  $t$ , RSU region  $i$  contains  $|r_i(t)|$  vehicle clusters. Each cluster  $j$  has a selected cluster-head CV  $v_{i,j,1}(t)$ , and the caching-node set contains the RSU and all selected CVs. The remote provider stores the complete repository of  $Q$  contents. Four service routes are modeled: same-cluster CV to requesting vehicle, RSU to requesting vehicle, another cluster's CV through the RSU, and remote backhaul retrieval. Figure 1 illustrates this architecture and the corresponding V2V, V2I, inter-cluster, and backhaul delivery paths.

Let  $N_i = |r_i(t)|$  denote the number of clusters in RSU region  $i$ ,  $\lambda_{i,j}(t)$  the request arrival rate of cluster  $j$ , and  $p_{q,i,j}(t)$  the probability that a request concerns content  $q$ . The probability

normalization and traffic-based cluster weights are defined in (1), while the aggregate RSU-level popularity is given by (2). LSTM predicts the next-period request counts for each zone/cluster; nonnegative predicted counts are normalized directly into probabilities rather than being reduced to rank information.

Let  $N_i = |r_i(t)|$  denote the number of clusters in RSU region  $i$ ,  $\lambda_{i,j}(t)$  the request arrival rate of cluster  $j$ , and  $p_{q,i,j}(t)$  the probability that a request concerns content  $q$ . The probability normalization and traffic-based cluster weights are defined in (1), while the aggregate RSU-level popularity is given by (2). LSTM predicts the next-period request counts for each zone/cluster; nonnegative predicted counts are normalized directly into probabilities rather than being reduced to rank information.

$$\sum_{q=1}^Q p_{q,i,j}(t) = 1, w_{i,j}(t) = \frac{\lambda_{i,j}(t)}{\sum_{k=1}^{N_i} \lambda_{i,k}(t)} \quad (1)$$

$$p_{q,i}(t) = \sum_{j=1}^{N_i} w_{i,j}(t) p_{q,i,j}(t) \quad (2)$$

The binary placement variable  $x_{q,n}(t)$  indicates whether content  $q$  of size  $s_q$  is stored at caching node  $n$ . The placement and capacity conditions are summarized in (3). Capacity constraints are imposed independently on CVs and RSUs. For direct comparison with CCCRP, strict non-redundancy can be enforced so that a content is stored at most once in an RSU region; controlled replication can be enabled by replacing this bound with a replication limit.

$$x_{q,n}(t) \in \{0,1\}, \sum_{q=1}^Q s_q x_{q,n}(t) \leq C_n \quad (3)$$

A second binary variable  $y_{q,i,j,n}(t)$  identifies the source that actually serves a request. As stated in (4), each request is assigned to exactly one edge source or to the backhaul, and an edge node may serve  $q$  only when it stores  $q$ . This explicit assignment is necessary because a cache hit may traverse different wireless paths.

$$\sum_n y_{q,i,j,n}(t) = 1, y_{q,i,j,n}(t) \leq x_{q,n}(t) \quad (4)$$

The expected cluster hit ratio  $H_{i,j}(t)$  is the probability that a request avoids the backhaul, and  $H_i(t)$  is the traffic-weighted regional hit ratio defined in (5). If  $d_{q,i,j,n}(t)$  is the acquisition delay from source  $n$ , the same weighting gives cluster delay  $D_{i,j}(t)$  and regional delay  $D_i(t)$  as given in (6).

$$H_{i,j}(t) = \sum_{q=1}^Q p_{q,i,j}(t) [1 - y_{q,i,j,BH}(t)], H_i(t) = \sum_{j=1}^{N_i} w_{i,j}(t) H_{i,j}(t) \quad (5)$$

$$D_{i,j}(t) = \sum_{q=1}^Q p_{q,i,j}(t) \sum_n y_{q,i,j,n}(t) d_{q,i,j,n}(t), D_i(t) = \sum_{j=1}^{N_i} w_{i,j}(t) D_{i,j}(t) \quad (6)$$

The joint utility in (7) balances the two aggregate metrics after delay normalization. A minimum cluster hit ratio  $H_{\min}$  is used to discourage solutions that achieve high aggregate performance while underserving a cluster. An optional cluster-delay bound can be added when a hard latency requirement is available.

$$J_i(t) = \alpha H_i(t) - (1 - \alpha) \frac{D_i(t)}{D_{\text{ref}}}, H_{i,j}(t) \geq H_{\min} \quad (7)$$

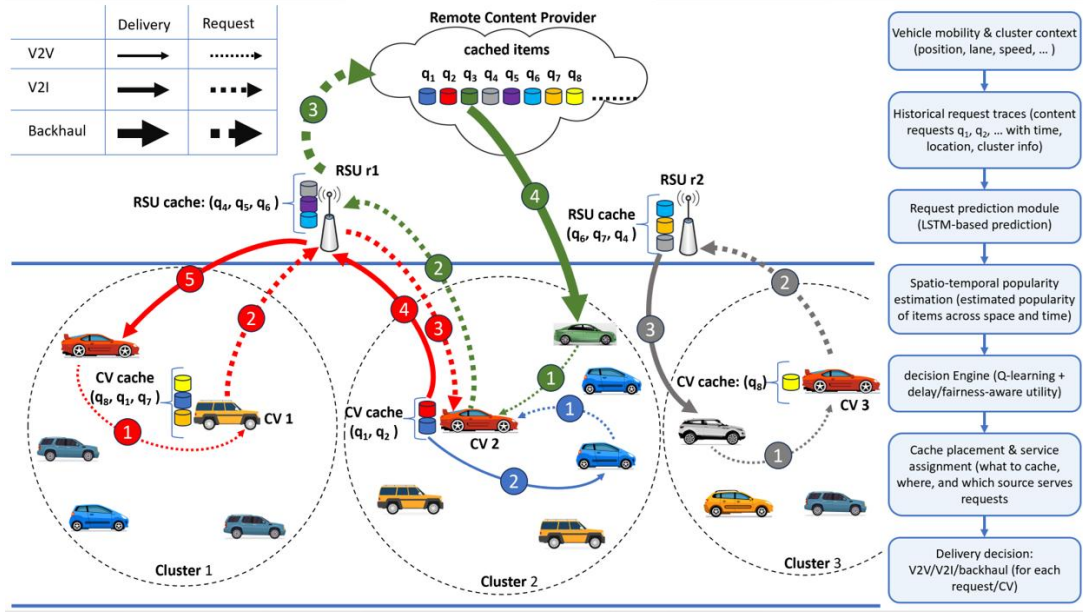


Figure 1: Illustrative network scenario for the proposed cooperative caching framework.

## 4. ST-3CRP Cooperative Caching Strategy

### 4.1. Vehicle Clustering and Cache-Update Context

At each cache-update epoch  $t$ , ST-3CRP first groups vehicles according to normalized position and speed and selects one caching vehicle (CV) for each cluster. The resulting cluster configuration determines the set of edge caching nodes available in RSU region  $i$  and provides the spatial context used by the prediction and service-assignment stages.

### 4.2. LSTM-Based Spatio-Temporal Request Prediction

For each content  $q$  in cluster  $j$  of RSU region  $i$ , the request-prediction module processes a fixed-length sequence of historical request counts covering the previous  $L$  cache-update periods. The LSTM maintains a hidden state  $h_t$  and a cell state  $c_t$ . Its input gate controls how much newly observed demand is written to memory, the forget gate controls how much previous demand information is retained, and the output gate determines the information exposed through the hidden state. The final hidden representation is mapped to the next-period predicted request count  $\widehat{x}_{q,i,j}(t+1)$ .

The prediction is conditioned on RSU-zone and cluster identity rather than being computed from a single network-wide sequence. Negative predicted counts are clipped to zero and the remaining values are normalized across the  $Q$  contents to obtain the cluster-specific request probabilities  $p_{q,i,j}(t)$ . This preserves both temporal evolution and spatial demand differences while retaining the magnitude information produced by the predictor.

### 4.3. Spatio-Temporal Popularity and Service-Path Evaluation

The predicted probabilities  $p_{q,i,j}(t)$  are combined with the cluster traffic weights  $w_{i,j}(t)$  to obtain regional popularity. For each feasible placement, the service-assignment variables  $y_{q,i,j,n}(t)$  identify whether a request is served by the local CV, the RSU, another cluster's CV through the RSU, or the backhaul. The corresponding path delay  $d_{q,i,j,n}(t)$  is therefore evaluated before the cache configuration receives a utility value. This step is essential because two placements may yield similar hit ratios while producing different acquisition delays and worst-cluster service levels.

#### 4.4. Q-Learning-Based Cache Placement

The Q-learning state summarizes available caching resources, predicted local-demand conditions, cluster configuration, and discretized delay conditions. An action  $a_k$  is a complete feasible cache-placement vector  $x(t)$ . For a selected action, service variables  $y$  are determined from cached availability and delivery delay. The reward is  $r_k = J_i(t) - \rho_k$ , where  $\rho_k$  penalizes service-constraint violations; when hard feasibility filtering is used,  $\rho_k$  can be zero. The Q-value update is given in (8), and exploration is controlled by an  $\epsilon$ -greedy policy.

$$Q(s_k, a_k) \leftarrow (1 - \eta)Q(s_k, a_k) + \eta[r_k + \gamma \max_{a'} Q(s_{k+1}, a')] \quad (8)$$

Q-learning is used as the cache-decision engine rather than as the request predictor. The LSTM supplies the estimated demand distribution, while the learning stage evaluates complete feasible placement matrices under the proposed hit-delay utility. For each selected placement, the service source of every cluster-content pair is obtained from the available cached copies and path delays before the reward is computed.

The formulation deliberately keeps the same Q-learning family as CCCRP so that the main methodological difference remains the objective and the spatio-temporal service model. Capacity, binary-placement, and replication constraints are enforced before a placement is evaluated, which prevents infeasible cache matrices from being rewarded. The final configuration is the feasible action with the highest learned value for the current demand context.

#### 4.5. Computational Considerations

A remaining limitation is the complete-placement action representation: the number of feasible matrices grows rapidly with the number of contents  $Q$  and caching nodes. ST-3CRP should therefore be interpreted as a formulation for moderate-size cooperative caching regions rather than as evidence of large-scale tabular-Q scalability. Hierarchical actions, factorized placement decisions, or approximate value functions are natural extensions for larger repositories and denser caching topologies.

### 5. Experimental Methodology

Two complementary simulation settings are used. The Reference Evaluation Setting follows the original CCCRP scale with 15 vehicles, 2 clusters,  $Q=10$  contents, 20 Mb nominal total cache capacity, default Zipf parameter  $\phi=0.6$ , spatial skew 0.85, request arrival rate 1 req/s, and 200 independent runs. The Extended Evaluation Setting uses  $Q=20$  contents, 4 clusters, 30 Mb total capacity,  $\phi=0.75$ , skew 0.85, cluster weights [0.45, 0.25, 0.175, 0.125], and 200 independent runs per operating point. LFU uses cumulative frequency, LRU uses the most recent requests, and CCCRP uses LSTM plus Zipf-based popularity.

The content-request trace used for prediction is the same dataset adopted in the original CCCRP study [6]. It contains request data for ten contents generated from the Big Data Challenge dataset reported by Barlacchi et al. [17], collected from November 1, 2013 to January 1, 2014 at 10-min intervals. Following the CCCRP protocol, 1000 chronological observations are retained: the first 800 are used to train the LSTM and the final 200 are used for testing.

The communication parameters are identical across strategies unless a sweep explicitly changes them. Content sizes are uniformly sampled from {1, 1.5, 2, 2.5, 3} Mb; bandwidth is 10 MHz; vehicle and RSU transmit powers are 25 and 50 dBm; thermal noise is -100 dBm; the path-loss exponent is 3.0; representative V2V and V2I distances are 80 and 300 m; and backhaul delay is 6 s. The objective weight is  $\alpha=0.6$ ,  $D_{\text{ref}}=3$  s, Q-learning learning rate  $\eta=0.5$ , and discount factor  $\gamma=0.9$ .

Table 1  
Principal simulation configuration s.

| Parameter                        | Reference setting | Extended setting |
|----------------------------------|-------------------|------------------|
| Vehicles                         | 15                | 40               |
| Clusters                         | 2                 | 4                |
| Contents Q                       | 10                | 20               |
| Nominal cache capacity           | 20 Mb             | 30 Mb            |
| Default Zipf parameter           | 0.60              | 0.75             |
| Spatial skew                     | 0.85              | 0.85             |
| Monte Carlo trials               | 200               | 200              |
| Q-learning ( $\eta$ , $\gamma$ ) | (0.5, 0.9)        | (0.5, 0.9)       |

For every operating point, 200 independent random seeds are evaluated. For a given seed, ST-3CRP, CCCRP, LFU, and LRU use the same environmental realization, including the request and network conditions, so that comparisons are paired and nuisance variability is reduced. Each curve reports the sample mean together with a 95% confidence interval. At prespecified key operating points, The statistical protocol uses paired Wilcoxon signed-rank tests between all strategies, followed by Holm correction across the selected tests.

In the reference clustering, the 15 vehicles are split into two clusters of nine and six vehicles, respectively. These clusters are formed in the normalized position-speed space, and each cluster features a distinct caching vehicle. The extended realization contains 40 vehicles distributed across four clusters of 16, 10, 9, and 5 vehicles, with one caching vehicle selected per cluster.

The extended experiments vary spatial skew, coupled cluster/repository/capacity scale, relative caching capacity  $\rho$ , Zipf parameter  $\phi$ , and request arrival rate. They report aggregate hit ratio and average delay, together with minimum cluster hit ratio and maximum cluster delay as worst-cluster service indicators. All reported sweep points are based on 200 paired runs and are accompanied by 95% confidence intervals.

## 6. Results and Discussion

### 6.1. Reference-Scale Validation and Baseline Comparison

The reference experiment first checks consistency with the original CCCRP results while adding uncertainty estimates from 200 independent paired runs. At  $\phi=0.7$ , Wang et al. reported hit ratios of approximately 88.3%, 83.4%, and 82.2% for CCCRP, LFU, and LRU, with delays of 1.34, 1.47, and 1.53 s [6]. The reproduced means are  $0.8831 \pm 0.0046$ ,  $0.8375 \pm 0.0045$ , and  $0.8221 \pm 0.0046$  for hit ratio, and  $1.3142 \pm 0.0142$ ,  $1.4616 \pm 0.0196$ , and  $1.5061 \pm 0.0112$  s for delay, where the  $\pm$  terms denote 95% CI half-widths. At the same point, ST-3CRP reaches  $0.8884 \pm 0.0023$  hit ratio and  $1.2767 \pm 0.0030$  s delay, corresponding to a 0.53-percentage-point hit-ratio gain and a 2.86% delay reduction relative to reproduced CCCRP. As shown in Figure 2, ST-3CRP maintains the best mean performance throughout the reference sweeps, while the error bars quantify the remaining run-to-run variability. Across  $\phi=0.5-0.9$ , its hit ratio increases from  $0.8737 \pm 0.0037$  to  $0.9072 \pm 0.0048$ , while delay decreases from  $1.3180 \pm 0.0074$  to  $1.2479 \pm 0.0012$  s. The capacity sweep produces a larger separation: at 5 Mb, ST-3CRP reaches  $0.3563 \pm 0.0159$  compared with  $0.3275 \pm 0.0150$  for CCCRP; at 20 Mb, the corresponding values are  $0.9343 \pm 0.0100$  and  $0.8922 \pm 0.0100$ .

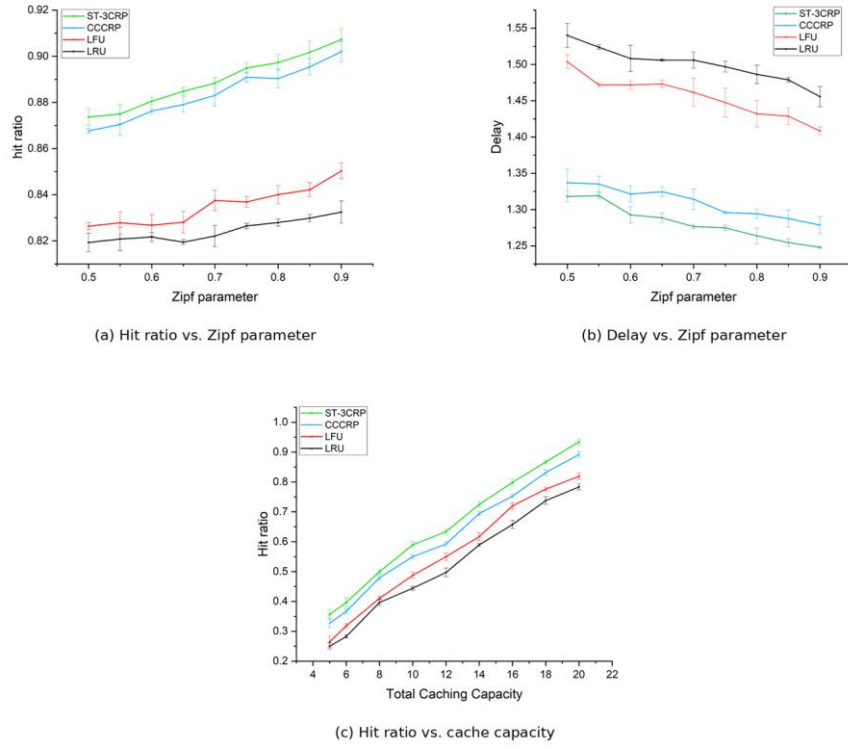


Figure 2: Performance under the Reference Evaluation Setting (mean with 95% confidence intervals): (a) hit ratio versus Zipf parameter, (b) delay versus Zipf parameter, and (c) hit ratio versus total cache capacity.

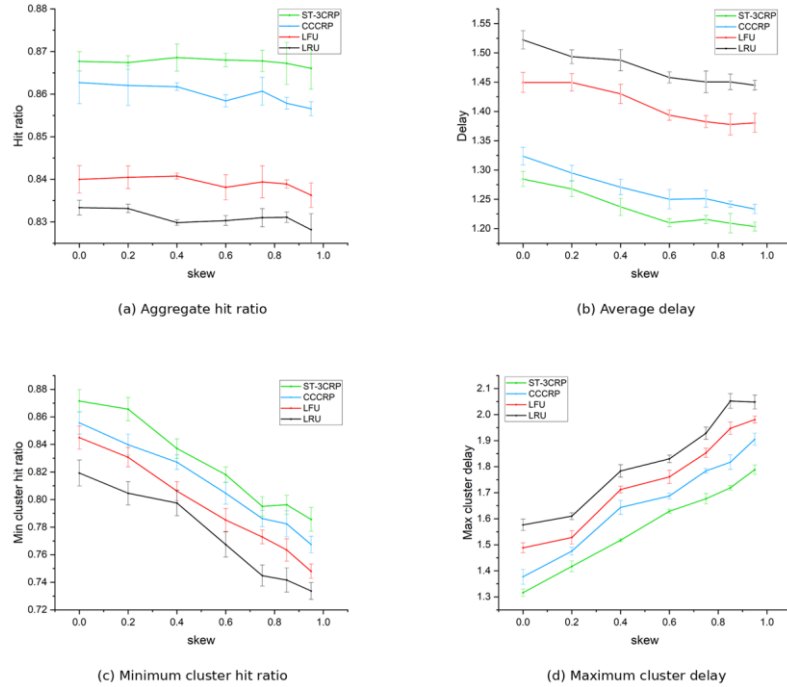


Figure 3: Effect of spatial demand skew (mean with 95% confidence intervals): aggregate hit ratio, average delay, minimum cluster hit ratio, and maximum cluster delay.

## 6.2. Spatial Imbalance and Fairness

As shown in Figure 3, spatial imbalance is a stronger test of the fairness-aware formulation because aggregate traffic weights can hide the least-favored cluster. As skew rises from 0 to 0.95, ST-3CRP aggregate hit ratio remains comparatively stable, changing from  $0.8677 \pm 0.0023$  to  $0.8661 \pm 0.0049$ , whereas its minimum cluster hit ratio decreases from  $0.8715 \pm 0.0082$  to  $0.7856 \pm 0.0086$  and maximum cluster delay increases from  $1.3159 \pm 0.0139$  to  $1.7886 \pm 0.0179$  s. At skew 0.95, ST-3CRP exceeds CCCRP by 1.82 percentage points in minimum cluster hit ratio and reduces maximum cluster delay by 6.14%; its average delay is also 2.42% lower than CCCRP.

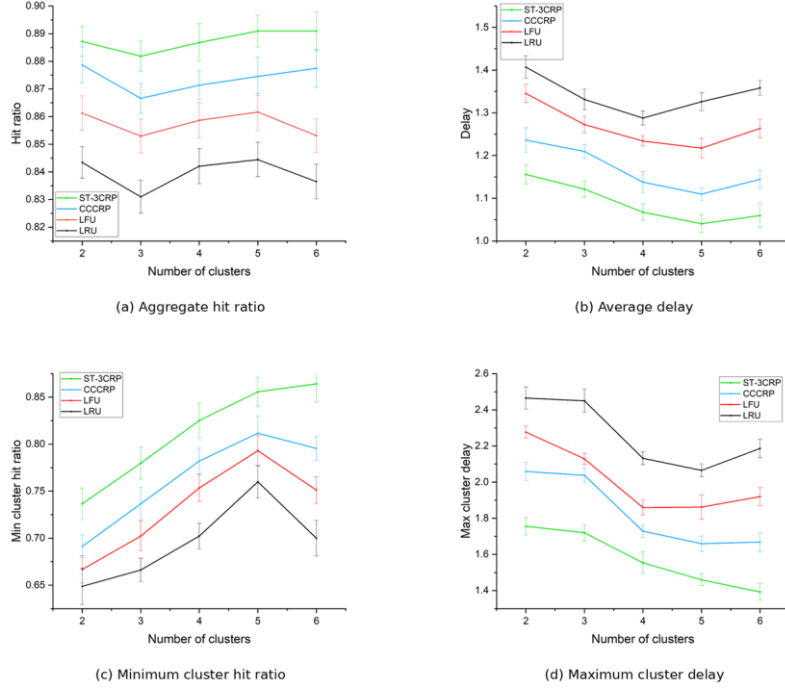


Figure 4: Performance under coupled scaling of cluster count, repository size, and total cache capacity (mean with 95% confidence intervals).

## 6.3. Coupled Cluster-Scaling Analysis

As summarized in Figure 4, the coupled scaling experiment changes the number of clusters from 2 to 6 while simultaneously increasing the repository from 13 to 27 contents and cache capacity from 20 to 40 Mb; it must therefore be interpreted as a joint scale change rather than an isolated cluster-count effect. ST-3CRP has the highest aggregate hit ratio and lowest average delay at every point. Relative to CCCRP, it improves minimum cluster hit ratio by 4.89 percentage points on average and reduces maximum cluster delay by 13.81% on average. At six clusters, ST-3CRP reaches a minimum cluster hit ratio of  $0.8642 \pm 0.0193$  versus  $0.7954 \pm 0.0126$  for CCCRP, while maximum cluster delay falls from  $1.6682 \pm 0.0535$  to  $1.3927 \pm 0.0468$  s, a 16.52% reduction.

## 6.4. Relative Cache-Capacity Analysis

As shown in Figure 5, cache scarcity produces the clearest aggregate gain. At  $\rho=0.25$  (10 Mb), ST-3CRP reaches hit ratio  $0.5615 \pm 0.0335$  versus  $0.4810 \pm 0.0240$  for CCCRP, an improvement of 8.04 percentage points, and reduces average delay from  $3.2098 \pm 0.0719$  to  $2.5669 \pm 0.0542$  s, a 20.03% reduction. At  $\rho=0.75$  (30 Mb), the hit-ratio gain is 6.30 percentage points and the average-delay reduction is 31.30%; minimum cluster hit ratio improves by 10.70 percentage points and maximum cluster delay falls by 14.66%. At  $\rho=1.20$ , hit ratios approach saturation, but ST-3CRP still reduces

average delay by 23.66%, showing that delay-aware placement remains useful even when additional hits are difficult to obtain.

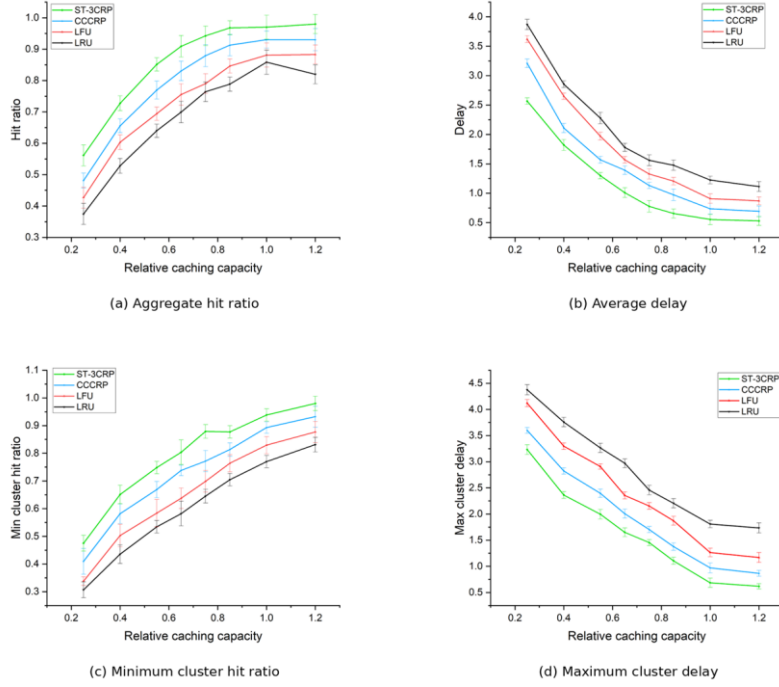


Figure 5: Effect of relative caching capacity  $\rho$  on aggregate and worst-cluster metrics (mean with 95% confidence intervals).

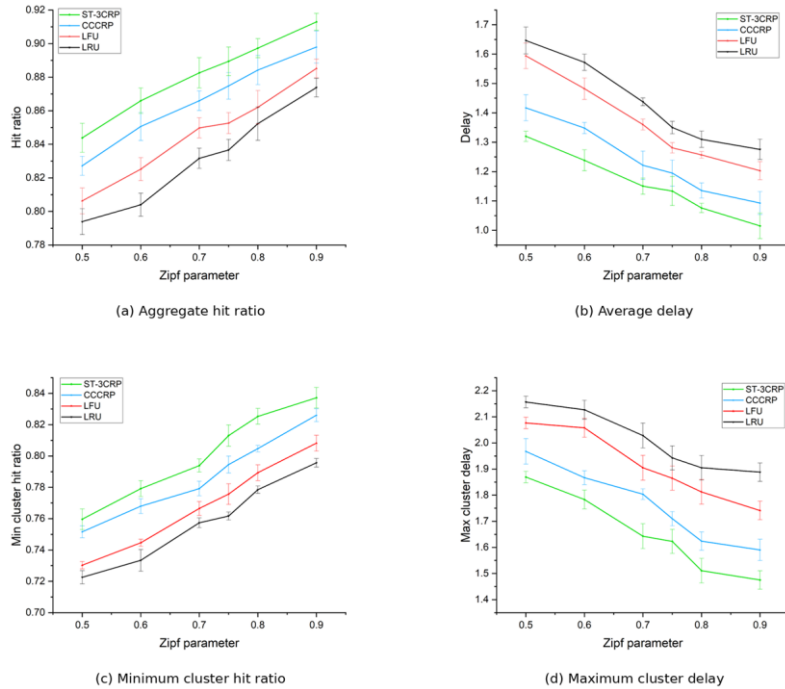


Figure 6: Effect of the Zipf parameter in the Extended Evaluation Setting (mean with 95% confidence intervals).

## 6.5. Zipf-Parameter Sensitivity

As shown in Figure 6, the extended Zipf sweep confirms the same behavior under changes in content concentration. From  $\varphi=0.5$  to  $0.9$ , ST-3CRP hit ratio increases from  $0.8438 \pm 0.0086$  to  $0.9129 \pm 0.0050$  and average delay decreases from  $1.3201 \pm 0.0175$  to  $1.0151 \pm 0.0435$  s. At  $\varphi=0.75$ , its minimum cluster hit ratio is  $0.8131$  versus  $0.7946$  for CCCRP, while maximum cluster delay is  $1.6230$  versus  $1.7098$  s. At  $\varphi=0.9$ , ST-3CRP improves aggregate hit ratio by  $1.50$  percentage points and reduces average delay and maximum cluster delay by  $7.12\%$  and  $7.25\%$ , respectively, relative to CCCRP.

## 6.6. Request-Arrival-Rate Sensitivity

As shown in Figure 7, request arrival rate is varied from  $0.5$  to  $3$  req/s. Average delay increases for all strategies, but ST-3CRP remains lowest. Its reduction relative to CCCRP is  $7.86\%$  at  $0.5$  req/s,  $9.92\%$  at  $1$  req/s, and  $10.67\%$  at  $3$  req/s; the average reduction over the sweep is  $8.80\%$ . Total-delay ordering also remains ST-3CRP, CCCRP, LFU, and LRU from lowest to highest, with an average reduction of  $2.47\%$  relative to CCCRP.

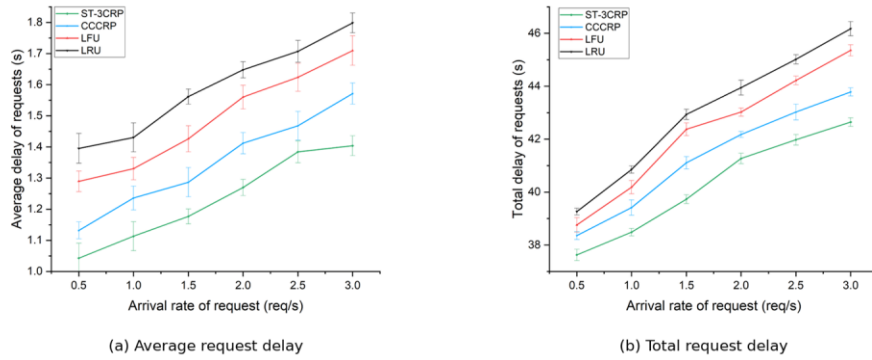


Figure 7: Effect of request arrival rate on average and total request delay (mean with 95% confidence intervals).

## 6.7. Statistical Uncertainty and Generalized Benchmark Summary

Table 2 consolidates representative operating points from the reference and extended evaluations and provides a compact benchmark across aggregate and worst-cluster metrics. Values are reported as mean  $\pm$  95% confidence-interval half-width over 200 paired runs. The added gain column expresses hit-ratio improvements in percentage points (pp) and delay improvements as relative reductions with respect to CCCRP.

Table 2

Generalized performance comparison at representative operating points. Values are mean  $\pm$  95% confidence-interval half-width over 200 paired runs; hit-ratio gains are in percentage points (pp), whereas delay gains are relative reductions versus CCCRP.

| Condition / metric                                       | ST-3CRP             | CCCRP               | LFU                 | LRU                 | Gain vs. CCCRP |
|----------------------------------------------------------|---------------------|---------------------|---------------------|---------------------|----------------|
| Reference, $\varphi=0.70$<br>Hit ratio $\uparrow$        | $0.8884 \pm 0.0023$ | $0.8831 \pm 0.0046$ | $0.8375 \pm 0.0045$ | $0.8221 \pm 0.0046$ | +0.53 pp       |
| Reference, $\varphi=0.70$<br>Avg. delay (s) $\downarrow$ | $1.2767 \pm 0.0030$ | $1.3142 \pm 0.0142$ | $1.4616 \pm 0.0196$ | $1.5061 \pm 0.0112$ | -2.86%         |

|                                                                                    |                        |                        |                        |                        |          |
|------------------------------------------------------------------------------------|------------------------|------------------------|------------------------|------------------------|----------|
| Extended, skew=0.95<br>Min. cluster hit $\uparrow$                                 | 0.7856<br>$\pm 0.0086$ | 0.7674<br>$\pm 0.0060$ | 0.7480<br>$\pm 0.0052$ | 0.7337<br>$\pm 0.0061$ | +1.82 pp |
| Extended, skew=0.95<br>Max. cluster delay (s) $\downarrow$                         | 1.7886<br>$\pm 0.0179$ | 1.9055<br>$\pm 0.0232$ | 1.9809<br>$\pm 0.0136$ | 2.0481<br>$\pm 0.0266$ | -6.14%   |
| Coupled scale: 6<br>clusters, Q=27, C=40 Mb<br>Min. cluster hit $\uparrow$         | 0.8642<br>$\pm 0.0193$ | 0.7954<br>$\pm 0.0126$ | 0.7511<br>$\pm 0.0143$ | 0.6999<br>$\pm 0.0190$ | +6.88 pp |
| Coupled scale: 6<br>clusters, Q=27, C=40 Mb<br>Max. cluster delay (s) $\downarrow$ | 1.3927<br>$\pm 0.0468$ | 1.6682<br>$\pm 0.0535$ | 1.9198<br>$\pm 0.0493$ | 2.1862<br>$\pm 0.0504$ | -16.52%  |
| Extended, $\rho=0.25$ (10<br>Mb)<br>Hit ratio $\uparrow$                           | 0.5615<br>$\pm 0.0335$ | 0.4810<br>$\pm 0.0240$ | 0.4269<br>$\pm 0.0335$ | 0.3745<br>$\pm 0.0335$ | +8.04 pp |
| Extended, $\rho=0.25$ (10<br>Mb)<br>Avg. delay (s) $\downarrow$                    | 2.5669<br>$\pm 0.0542$ | 3.2098<br>$\pm 0.0719$ | 3.6196<br>$\pm 0.0583$ | 3.8708<br>$\pm 0.0871$ | -20.03%  |
| Extended, $\varphi=0.90$<br>Hit ratio $\uparrow$                                   | 0.9129<br>$\pm 0.0050$ | 0.8979<br>$\pm 0.0096$ | 0.8852<br>$\pm 0.0055$ | 0.8738<br>$\pm 0.0055$ | +1.50 pp |
| Extended, $\varphi=0.90$<br>Avg. delay (s) $\downarrow$                            | 1.0151<br>$\pm 0.0435$ | 1.0928<br>$\pm 0.0390$ | 1.2029<br>$\pm 0.0307$ | 1.2755<br>$\pm 0.0347$ | -7.12%   |
| Extended, arrival=3<br>req/s<br>Avg. delay (s) $\downarrow$                        | 1.4038<br>$\pm 0.0317$ | 1.5715<br>$\pm 0.0338$ | 1.7095<br>$\pm 0.0473$ | 1.7985<br>$\pm 0.0317$ | -10.67%  |

## 6.8. Overall Discussion

Overall, the strongest evidence is obtained against CCCRP because both predictive methods share the LSTM/Q-learning family and are evaluated under the same paired environmental realizations. The 200-run confidence intervals show that the ranking of the mean curves is stable across the tested operating points. Aggregate hit-ratio differences remain modest in high-hit regimes, but delay and worst-cluster metrics reveal larger benefits from conditioning demand spatially and accounting for heterogeneous service paths. The gains become especially pronounced under cache scarcity and coupled scaling, where both fairness and delay margins widen. The calibrated fairness target  $H_{min}$  should nevertheless be interpreted as an operational target rather than a universally guaranteed hard constraint.

Scope of the evaluation. The present study focuses on the end-to-end impact of spatio-temporal demand conditioning, delay-aware placement, and cluster-level service balancing. The experiments provide statistical uncertainty estimates through 200 independent paired runs and 95% confidence intervals. Component-level prediction assessment and computational scaling of the complete-placement Q-learning action representation are outside the present experimental scope and constitute natural extensions of this work.

## 7. Conclusion

ST-3CRP extends predictive cooperative caching by jointly modeling where future requests arise, which cache stores the requested content, and which communication path serves it. The evaluation uses 200 independent paired runs per operating point and 95% confidence intervals. The reference setting remains consistent with the behavior of the original CCCRP baseline, while the extended setting shows larger benefits under cache scarcity and heterogeneous cluster service. At  $p=0.25$ , ST-3CRP improves hit ratio over CCCRP by 8.04 percentage points and reduces average delay by 20.03%. Across coupled scale changes, it improves minimum cluster hit ratio by 4.89 percentage points on average and reduces maximum cluster delay by 13.81% on average. These results indicate that the primary advantage of ST-3CRP is not only additional cache hits, but also lower latency and more balanced use of those hits. Future work will extend the present evaluation toward independent prediction-quality assessment, scalable cache-action representations, and explicit mobility/contact-aware placement under larger and more dynamic vehicular scenarios.

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (OpenAI) for language refinement, manuscript organization, and formatting assistance. The authors reviewed and edited the resulting content and take full responsibility for the publication's content.

## References

- [1] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, *IEEE Internet of Things Journal* 3 (2016) 637-646. doi:10.1109/JIOT.2016.2579198.
- [2] Y. Mao, C. You, J. Zhang, K. Huang, K. B. Letaief, A survey on mobile edge computing: The communication perspective, *IEEE Communications Surveys & Tutorials* 19 (2017) 2322-2358. doi:10.1109/COMST.2017.2745201.
- [3] L. Hou, M. A. Gregory, S. Li, A survey of multi-access edge computing and vehicular networking, *IEEE Access* 10 (2022) 123436-123451. doi:10.1109/ACCESS.2022.3224032.
- [4] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, D. Kutscher, Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions, in: *Proceedings of the Workshop on Mobile Edge Communications (MECOMM)*, 2017, pp. 7-12. doi:10.1145/3098208.3098210.
- [5] G. Paschos, E. Bastug, I. Land, G. Caire, M. Debbah, Wireless caching: Technical misconceptions and business barriers, *IEEE Communications Magazine* 54 (2016) 16-22. doi:10.1109/MCOM.2016.7537172.
- [6] R. Wang, Z. Kan, Y. Cui, D. Wu, Y. Zhen, Cooperative caching strategy with content request prediction in Internet of Vehicles, *IEEE Internet of Things Journal* 8 (2021) 8964-8975. doi:10.1109/JIOT.2021.3056084.
- [7] L. Yao, A. Chen, J. Deng, J. Wang, G. Wu, A cooperative caching scheme based on mobility prediction in vehicular content centric networks, *IEEE Transactions on Vehicular Technology* 67 (2018) 5435-5444. doi:10.1109/TVT.2017.2784562.
- [8] Z. Su, Y. Hui, Q. Xu, T. Yang, J. Liu, Y. Jia, An edge caching scheme to distribute content in vehicular networks, *IEEE Transactions on Vehicular Technology* 67 (2018) 5346-5356. doi:10.1109/TVT.2018.2824345.

- [9] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, L. Zhu, A mobility-aware vehicular caching scheme in content centric networks: Model and optimization, *IEEE Transactions on Vehicular Technology* 68 (2019) 3100-3112. doi:10.1109/TVT.2019.2899923.
- [10] M. Chen, W. Saad, C. Yin, M. Debbah, Echo state networks for proactive caching in cloud-based radio access networks with mobile users, *IEEE Transactions on Wireless Communications* 16 (2017) 3520-3535. doi:10.1109/TWC.2017.2683482.
- [11] L. Li et al., Deep reinforcement learning approaches for content caching in cache-enabled D2D networks, *IEEE Internet of Things Journal* 7 (2020) 544-557. doi:10.1109/JIOT.2019.2951509.
- [12] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, N. Ansari, Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks, *IEEE Internet of Things Journal* 7 (2020) 247-257. doi:10.1109/JIOT.2019.2945640.
- [13] Z. Nan, Y. Jia, Z. Ren, Z. Chen, L. Liang, Delay-aware content delivery with deep reinforcement learning in Internet of Vehicles, *IEEE Transactions on Intelligent Transportation Systems* 23 (2022) 8918-8929. doi:10.1109/TITS.2021.3087833.
- [14] Q. Wang, D. Grace, Proactive edge caching in vehicular networks: An online bandit learning approach, *IEEE Access* 10 (2022) 131246-131263. doi:10.1109/ACCESS.2022.3229645.
- [15] S. Khanal, K. Thar, E.-N. Huh, Route-based proactive content caching using self-attention in hierarchical federated learning, *IEEE Access* 10 (2022) 29514-29527. doi:10.1109/ACCESS.2022.3157637.
- [16] N. Aung, S. Dhelim, L. Chen, A. Lakas, W. Zhang, H. Ning, S. Chaib, M. T. Kechadi, VeSoNet: Traffic-aware content caching for vehicular social networks using deep reinforcement learning, *IEEE Transactions on Intelligent Transportation Systems* 24 (2023) 8638-8649. doi:10.1109/TITS.2023.3250320.
- [17] G. Barlacchi et al., A multi-source dataset of urban life in the city of Milan and the Province of Trentino, *Scientific Data* 2 (2015) 150055.
- [18] W. Tian, Y. Chen, F. Ke, H. Song, Deep reinforcement learning-based mobile-aware edge cooperative caching scheme in the Internet of Vehicles, *IEEE Transactions on Vehicular Technology* 74 (2025) 5053-5068. doi:10.1109/TVT.2024.3501364.
- [19] Y. Chen, W. Tian, Z. Li, H. Song, A cooperative caching scheme utilizing regional feature and dynamic vehicle clustering in vehicular edge networks, *Vehicular Communications* 53 (2025) 100912. doi:10.1016/j.vehcom.2025.100912.
- [20] L. Zhu, B. Li, L. Tan, Vehicular edge cloud computing content caching optimization solution based on content prediction and deep reinforcement learning, *Ad Hoc Networks* 165 (2024) 103643. doi:10.1016/j.adhoc.2024.103643.
- [21] M. A. Ghamri, B. Djamaa, M. A. Benatia, R. Bellahmer, An optimized Multi Agent Reinforcement Learning solution for edge caching in the Internet of Vehicles, *Pervasive and Mobile Computing* 112 (2025) 102081. doi:10.1016/j.pmcj.2025.102081.

- [22] Z. Cheng, L. Liu, M. Liwang, N. Chen, X. Fan, Learning-based joint recommendation, caching, and transmission optimization for cooperative edge video caching in Internet of Vehicles, *Ad Hoc Networks* 166 (2025) 103667. doi:10.1016/j.adhoc.2024.103667.