Collaborative Deep Q-Learning-Based Coded Data Uploading in Vehicular Micro Clouds

Ziqi Zhou*, Agon Memedi*, Chunghan Lee†, Onur Altintas†, Falko Dressler*
* School of Electrical Engineering and Computer Science, TU Berlin, Germany
† InfoTech Labs, Toyota Motor North America R&D, CA, U.S.A.
Email: {zhou, memedi, dressler}@ccs-labs.org, {chunghan.lee1, onur.altintas}@toyota.com

***Abstract*—Nowadays, electric vehicles (EVs) generate substantial data that is time-tolerant but costly to transmit via cellular networks. The architecture leverages standardized vehicle-to-vehicle (V2V) communication in combination with recent achievements towards virtualized edge computing in the vehicular environment. The idea is to offload data to cars that have a shorter time-to-charge for cost-quality-efficient, collaborative data uploading; we assume that charging points provide efficient internet connectivity in the near future. We formulate a two-stage architecture aiming to minimize monetary costs while maximizing data recovery rates at the data center. Our solution integrates data encoding schemes (e.g., Reed-Solomon codes (RS)) with deep Q-learning (DQL)-based distribution strategies within vehicular micro clouds (VMCs). For evaluation, we use close-to-real charging patterns of EVs. Simulations using realistic urban mobility patterns demonstrate that the combination of RS and DQL-based distribution significantly reduces cellular costs while maintaining high data recovery rates.**

***Index Terms*—Edge computing, vehicular micro cloud, collaborative uploading, deep Q-learning, reinforcement learning**

I. INTRODUCTION

The standardization of vehicle-to-vehicle (V2V) communication technologies acts as many new applications ranging from collaborative perception to coordination of driving maneuvers [1, 2]. In this context, the concept of vehicular micro clouds (VMCs) has been proposed [3], where vehicles jointly use available ICT resources in the form of a virtual edge server. Within a VMC, vehicles can communicate and assist each other offloading tasks and also providing distributed data storage.

In the scope of this paper, we are interested in using distributed storage for collaborative data upload to a backend cloud system. In practice, connected electric vehicles (EVs) generate, or record data during driving, which might be required for long-term storage and analysis. While some of this data is time-tolerant, deadlines in the order of minutes to hours may apply. The large scale of this data presents a significant burden, particularly if all of data is uploaded via cellular networks. Uploading data while charging may be a cost-efficient opportunity: while charging, cars are assumed to have access to free connections via WiFi or even fiber connections in the charger. Consequently, data collected by an EV can reach the data center via two communication paths: First, data may be forwarded to another EV in the same VMC via V2V links for intermediate storage and upload during charge time. Secondly, data will be uploaded via the cellular network.

In this paper, we present a deep Q-learning (DQL)-based approach in combination with coding to optimize the upload while ensuing timely upload within the given deadlines. We target two primary objectives: (1) minimizing the monetary cost by using fewer cellular links; (2) enhancing the data recovery rate at the data center by effectively encoding data and intelligently distributing encoded data in VMCs.

Our main contributions are summarized as follows:

- We formulate a two-stage framework for data uploading that integrates heterogeneous communication technologies, including V2V, WLAN, fiber, and cellular networks.
- We propose a novel combination of data encoding and data distribution strategies in VMCs to enable real-time decisions on data forwarding and uploading.
- We evaluate the proposed approach using an area in realistic city-scale scenarios simulated in SUMO and incorporating close-to-realistic EV charging patterns.
- We achieve an adaptive and well-balanced trade-off between monetary cost and data recovery quality.

II. RELATED WORK

Collaborative uploading: In heterogeneous networks, data is split into chunks and distributed across multiple transmission paths to optimize performance and reduce waiting time [4]. In mobile crowd sensing, researchers have focused on minimizing the cost of uploading by leveraging edge computing and the redundant resources of nearby nodes Xu and Song [5]. In the vehicular domain, Lee et al. [6] provide a direct foundation for our work by investigating the feasibility of collaborative data uploading in urban Los Angeles. Their analysis of vehicle roles in uploading frameworks offers critical insights into the practical conditions necessary for cooperative schemes.

Learning-based distribution strategies: Making optimal real-time decisions in dynamic vehicular environments is a complex challenge, increasingly addressed with learning-based methods. Huang et al. [7] proposed a deep reinforcement learning (DRL)-based online offloading framework to make offloading decisions in wireless powered edge computing networks, achieving near-optimal performance with low computational latency. Liu et al. [8] explored a vehicular edge network, where vehicles act as mobile servers and employed both Q-learning-based and DRL methods to find optimal policies, solving the weakness of no service in internet of things (IoT) networks caused by only static edge servers.

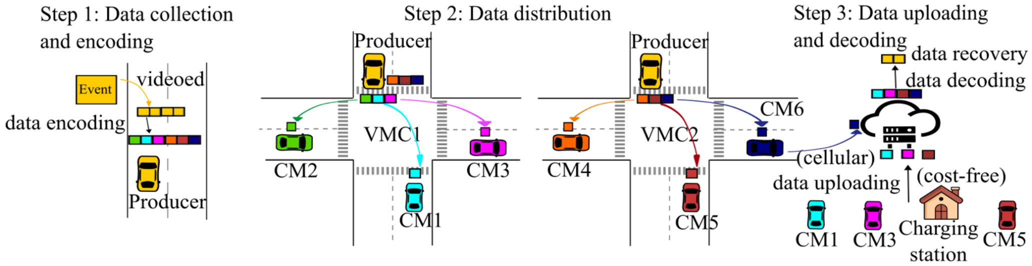


Figure 1. System model: From data collection to distribution to uploading

These works collectively validate DRL as a powerful tool for decision-making in edge environments. However, they primarily focus on computation offloading and resource splitting. Our work uniquely adapts and extends these principles to the distinct problem of cooperative data uploading in VMCs, introducing a novel DQL-based strategy that optimizes the trade-off between monetary cost and data recovery rate by intelligently distributing encoded data chunks.

Coding for data recovery: For improved data recovery, duplication is often used for simplicity, but certainly requiring more networking resources. Network coding like XOR-based schemes, helps reduce this overhead. Recently, erasure coding, particularly Reed-Solomon codes (RS) coding, is widely adopted in distributed systems for improved fault tolerance with lower redundancy. Bocharova et al. [9] demonstrated that RS codes offer better performance than baselines in city traffic scenarios, although with higher computational complexity.

III. COLLABORATIVE DATA UPLOADING

A. System Model

Vehicles are categorized into two roles: data producers that collect of generate data, encode the data as required (Step 1 in Figure 1), and those that collaboratively help temporarily storing and uploading data to the data center. Producers may upload their own data directly or forward it to other cars, i.e., cloud members (CMs), but they do not receive data from other vehicles. All cars form a stationary VMC [3] in which they communicate and distribute encoded data via V2V communication (Step 2 in Figure 1). Each data element is associated with a deadline. Thus, the vehicle holding the element must decide when and through which technology to upload. If a vehicle is scheduled to charge before the deadline, it stores it until charging, where we assume a free of charge connection (e.g., WiFi or fiber-connection) is available (Step 3 in Figure 1). Otherwise, the vehicle uploads the data via a cellular, incurring some monetary cost.

B. Central Controller and Data Center

A controller in each VMC coordinates data distribution within its area. It initializes learning models (if needed), collects vehicle information (e.g., position, charging information), and configures encoding at data producers. Based on the distribution strategy (we study random, greedy, and DQL), it selects an optimal CM as receiver or keeps the chunk with the generator. The data center monitors data recovery; once a data item is recoverable, it informs vehicles to cancel pending uploads of chunks belonging to the same data, avoiding redundant uploads. However, each vehicle should decide whether to upload immediately (cellular) or wait until charging (free cost), using its local knowledge of its charging schedule and chunk deadlines.

Table I
ENCODING SCHEME CHARACTERISTICS

Scheme	Chunks	Redundancy	Recovery
NON	10 raw	None	All 10 needed
DUP	20 raw	2× replication	Any 10 unique
XOR	10+2 parity	XOR encoding	5 or 4+parity
RS	14 coded	Non-systematic RS	Any 10 of 14

C. Encoding Schemes

Our system employs application-layer encoding on fixed-size data chunks. Each data item of size 30 MB is partitioned into $k = 10$ chunks of size 3 MB. Additional chunks are generated by encoding schemes for redundancy:

- Non-coding (NON): No additional chunks. Data remains partitioned into 10 chunks, with a total size of 30 MB.
- Duplication (DUP): Each chunk is replicated, doubling the total chunk number to $n = 20$ and total size to 60 MB.
- XOR parity (XOR): Groups of 5 raw chunks produce 1 parity chunk via bit-wise XOR: $p = r_1 \oplus r_2 \oplus \cdots \oplus r_5$. The scheme yields 12 chunks in total, introducing 20 % storage overhead (36 MB, $n = 12$).
- Reed-Solomon (RS): A $(n = 14, k = 10)$ non-systematic code generates 14 coded chunks. This adds 40 % storage overhead, resulting in 42 MB overall.

Each encoding scheme presents a distinct trade-off between storage and communication cost, recovery probability at the data center, and exposure rate at the receiver cars.

D. Distribution and Uploading Strategies

To efficiently distribute encoded data chunks in VMCs, we implement three distribution strategies, selecting which CM should receive a forwarded chunk. Before utilizing any distribution strategies, the producer always checks if its own charging schedules can meet the data deadline.

Random: As a baseline, in the random strategy, the controller selects a CM randomly from those within the same VMC. This approach requires no additional information, but also does not optimize for the charging pattern.

Table II
ROAD TRAFFIC PARAMETERS IN NAGOYA SCENARIO

Time	**Avg. # of producers**	**Avg. dwell time**
Morning	127	241.63 s
Afternoon	272	271.02 s
Evening	15	218.76 s

Greedy: The greedy strategy prioritizes local optimization. For each chunk, the controller selects the CM within the same VMC that will be charged earliest. Formally, for a chunk c generated by p, the algorithm evaluates all of p's VMC members $m \in \mathcal{M}_p$ and chooses:

$$m^* = \underset{m \in \mathcal{M}_p}{\operatorname{argmin}} \left(\max(0, t_{\text{charging}}(m) - t_{\text{now}}) \right),$$

where $t_{\text{Charging}}(m)$ is the car's charging time.

DQL: We employ deep Q-learning to address the chunk distribution problem, leveraging a deep neural network to approximate the Q-value function. This enables the learning agent (in the controller) to learn optimal actions within observed environments characterized by the state spaces. Our state space integrates vehicle features, VMC context, and pending chunk information. The action space contains selectable CMs within the same VMC. For Q-function approximation, we utilize a deep neural network trained with experience replay and target network stabilization techniques. The reward design encourages upload successes at the charger, penalizes cellular usage, and provides additional rewards for successful forwarding actions.

E. Upload Decision

Once a chunk is stored on a vehicle (either the original producer or a CM), the vehicle must decide when and how to upload it to the data center. If the vehicle's next charging opportunity t_{charging} occurs before the chunk's deadline t_{deadline}, the upload is deferred until t_{charging}, using the free connection available during charging. Otherwise, the chunk is uploaded immediately via the cellular network.

IV. EVALUATION

A. Simulation Parameters

For evaluation, we selected the Nagoya urban mobility (NUMo) scenario [10], which offers one of the most comprehensive urban traffic datasets, capturing highly realistic vehicle mobility across the entire city of Nagoya. We evaluate our approach during three distinct periods: morning (medium traffic), afternoon (high density), and late evening (low traffic), across nine VMCs. Each VMC corresponds to a major intersection covering approximately $100\,\text{m} \times 100\,\text{m}$. The scenarios differ in the number of vehicles per time step and in different dwell times (the time a vehicle spends in the VMC) within each area. Relevant traffic parameters for the selected VMCs are provided in Table II. To examine system performance under more realistic conditions, we configure a set of scenarios that vary both vehicular traffic and vehicular charging patterns. Vehicle charging behavior follows the close-to-real-world patterns described in [11]. All traffic scenarios are simulated using 1 s time step. We use data size 30 MB [6], a cellular price of 0.01 \$/MB, and a uniformly distributed deadline for each data item within the next 24 h.

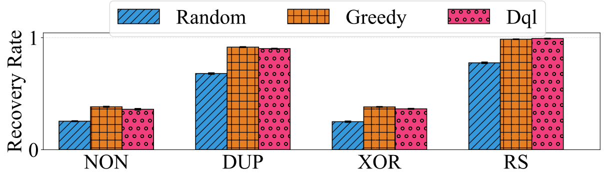


Figure 2. Data recovery rate in the morning Nagoya scenario

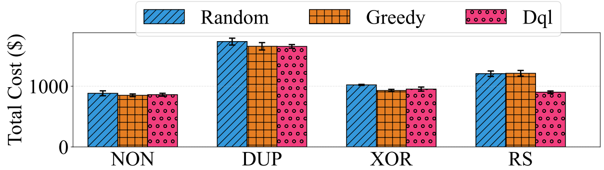


Figure 3. Upload cellular cost in the morning Nagoya scenario

For V2V communication between connected electric vehicles within the same VMC, we adopt the throughput model by Lee et al. [12], where the achievable data rate depends on inter-vehicle distance. We further set a 5 % failure probability for both V2V transmissions and subsequent uploads (regardless of the uploading technology). Based on the findings reported in [6], a VMC consisting of 25 % forwarder vehicles is sufficient to achieve a high data uploading ratio. Accordingly, we adopt a similar composition in our experiments: 75% of vehicles are designated as producers responsible for collecting and forwarding (or uploading their own) data, while the remaining 25% serve as potential forwarders (CMs) that receive and upload the forwarded data to the data center.

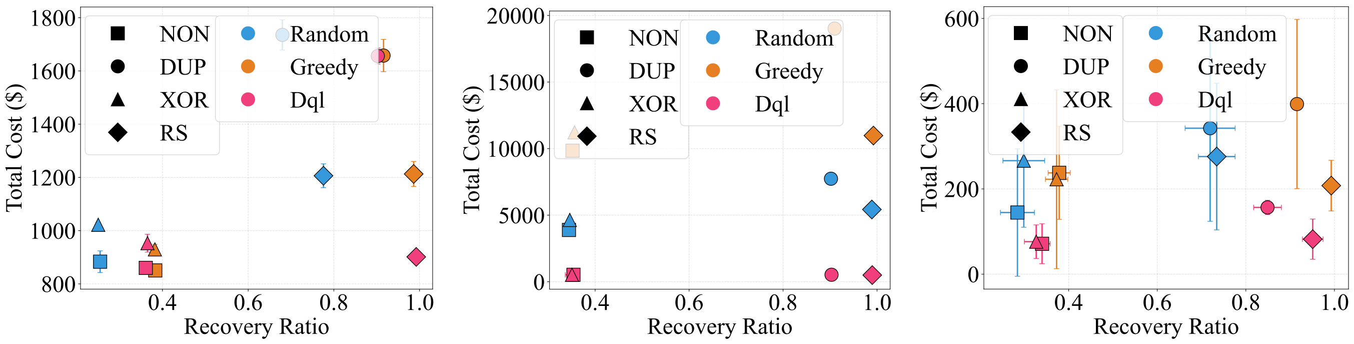
B. Recovery Rate and Total Cost

Without loss of generality, we present the results for the medium-density morning scenario as a representative example. Figure 2 shows the percentage of data that can be successfully recovered at the data center after all necessary uploads complete (all error bars in this paper represent standard deviations). Greedy with RS and our proposed DQL with RS outperform other combinations, while nearly all distribution strategies paired with RS achieve higher recovery rates. DUP also performs reasonably well, whereas NON and XOR obtain notably lower recovery rates (around 35 %). Figure 3 illustrates the monetary cost incurred by each encoding-distribution pair. DUP typically incurs the highest cost, while the other three schemes exhibit lower and comparable expenditures.

C. Recovery-Rate-Cost Trade-Off

Figure 4 offers an intuitive visualization of the trade-off between expenditure and recovery performance. Those in the lower-right corner represent combinations that spend less money while achieving higher recovery rates. The figure covers three scenarios differing in the average number of vehicles in the scenario (cf. Table II).

In the medium density traffic scenario (morning), as shown in Figure 4a, the combination of RS with DQL occupies the lower-right corner of the plot, achieving the best balance between



(a) Morning scenario

(b) Afternoon scenario

(c) Evening scenario

Figure 4. Cost vs. recovery comparison in the Nagoya scenario (morning vs. afternoon vs. evening)

recovery quality and cellular cost in the least dense traffic morning scenario. RS paired with Greedy also performs well relative to the rest of the methods, despite being slightly worse than the optimal trade-off in either cost or recovery rate.

In the high-density traffic scenario (afternoon), Figure 4b shows that RS-DQL yields an optimal compromise between monetary cost and recovery rate. RS-based methods attain high recovery rates, while DQL-based methods maintain low financial cost in the afternoon. For the low-density traffic scenario (evening) in Figure 4c, the most notable feature is the larger error bars (reflecting higher standard deviation), likely resulting from fewer vehicles in the VMCs. RS-DQL again achieves the most satisfactory trade-off, positioned at the lower-right corner with high recovery and low cost. RS-Greedy attains nearly 100% recovery, though at a moderately higher expense.

In all cases, the RS–DQL combination consistently outperforms the alternatives, achieving a favorable trade-off between cost and service quality.

V. CONCLUSION

We proposed a comprehensive framework that combines data encoding with learning-based distribution strategies to address the collaborative data uploading problem. The core idea is to forward encoded data chunks in the same VMC to vehicles likely to have earlier cost-free upload opportunities during EV charging. Our evaluation using realistic SUMO simulations of urban traffic and EV charging patterns demonstrates the effectiveness of the approach. Among the encoding schemes, RS coding consistently provided the best balance of monetary burden and recovery robustness. When paired with advanced distribution strategies, our proposed RS-DQL combination algorithm achieves superior performance, minimizing cellular upload costs while maximizing the data recovery rate at the data center. This work reveals the potential benefits of integrating erasure coding with reinforcement learning (RL) for decision-making in dynamic vehicular networks. Future work may explore more complex encoding adaptations, multi-VMC interactions, and the integration of predictive models for urban charging station availability to further enhance system performance and scalability.

Based on the potential of combining erasure coding with RL, future work could focus on an automatic and adaptive encoding-scheme selection protocol, which can dynamically adjust to varying environmental conditions, such as traffic patterns, data density, and data types. Moreover, charging-pattern prediction models represent another promising future. This will help estimating the time-to-charge and thus optimizing the system for large-scale city dynamics.

REFERENCES

[1] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz, “Inter-Vehicle Communication – Quo Vadis,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 170–177, Jun. 2014.

[2] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, “Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2377–2396, Jun. 2015.

[3] F. Dressler, G. S. Pannu, F. Hagenauer, M. Gerla, T. Higuchi, and O. Altintas, “Virtual Edge Computing Using Vehicular Micro Clouds,” in *IEEE International Conference on Computing, Networking and Communications (ICNC 2019)*, Honolulu, HI, Feb. 2019.

[4] W. R. KhudaBukhsh, B. Alt, S. Kar, A. Rizk, and H. Koeppl, “Collaborative Uploading in Heterogeneous Networks: Optimal and Adaptive Strategies,” in *37th IEEE Conference on Computer Communications (INFOCOM 2018)*, Honolulu, HI, Apr. 2018, pp. 1–9.

[5] C. Xu and W. Song, “Efficient Data Uploading for Mobile Crowdsensing via Team Collaborating and Matching,” *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 645–654, Mar. 2022.

[6] C. Lee, T. Higuchi, S. Ucar, N. Kaneko, O. Altintas, and K. Oguchi, “Is Collaborative Data Uploading Feasible? A Case for Los Angeles with Vehicular Micro Clouds,” in *100th IEEE Vehicular Technology Conference (VTC2024-Fall)*, Washington, D.C., Oct. 2024, pp. 1–5.

[7] L. Huang, S. Bi, and Y.-J. A. Zhang, “Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[8] Y. Liu, H. Yu, S. Xie, and Y. Zhang, “Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 158–11 168, Nov. 2019.

[9] I. Bocharova, B. Kudryashov, N. Lyamin, E. Frick, M. Rabi, and A. Vinel, “Low Delay Inter-Packet Coding in Vehicular Networks,” *Future Internet*, vol. 11, no. 10, May 2019.

[10] T. Higuchi, L. Zhong, and R. Onishi, “NUMo: Nagoya Urban Mobility Scenario for City-Scale V2X Simulations,” in *15th IEEE Vehicular Networking Conference (VNC 2024)*, Kobe, Japan, May 2024, pp. 17–24.

[11] N. Gholizadeh, *Electric Vehicle Charging Dataset*, version V1, 2024. [Online]. Available: http://dx.doi.org/10.17632/5zrtmp7gwd.1.

[12] C. Lee, S. Ucar, O. Altintas, and K. Oguchi, “Performance Characteristics of TCP and QUIC over V2V Wi-Fi in Urban Scenarios,” in *43rd IEEE Military Communications Conference (MILCOM 2025)*, Los Angeles, CA, Oct. 2025, pp. 662–667.