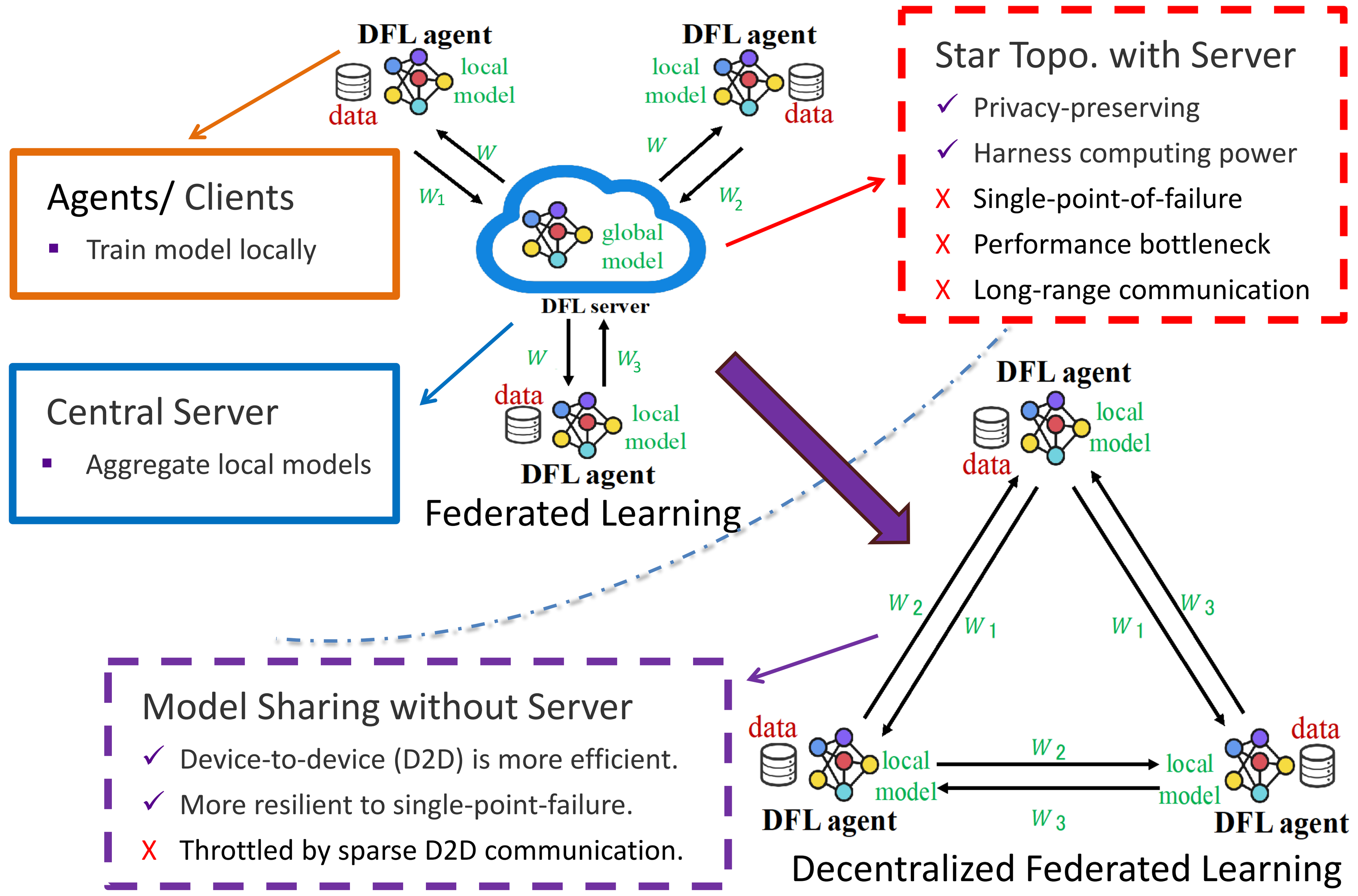
Decentralized Federated Learning with Model Caching on Mobile Agents

AAAI-25 / IAAI-25 / EAAI-25
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, USA

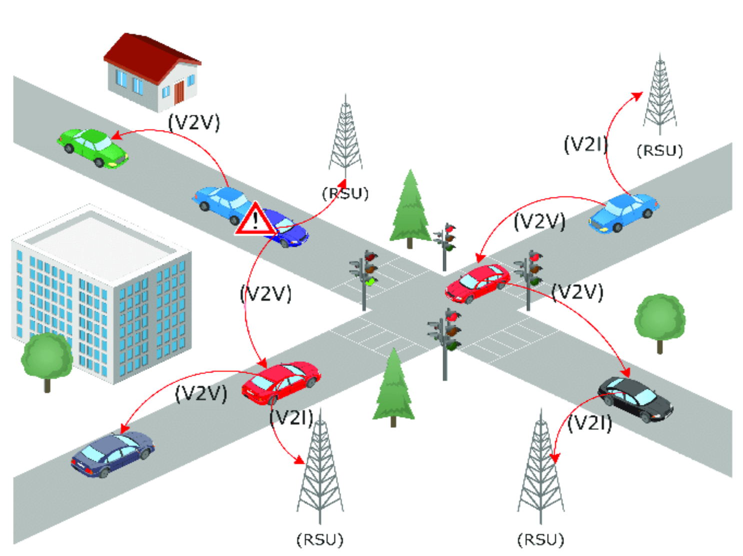
Code: https://github.com/ShawnXiaoyuWang/Cached-DFL

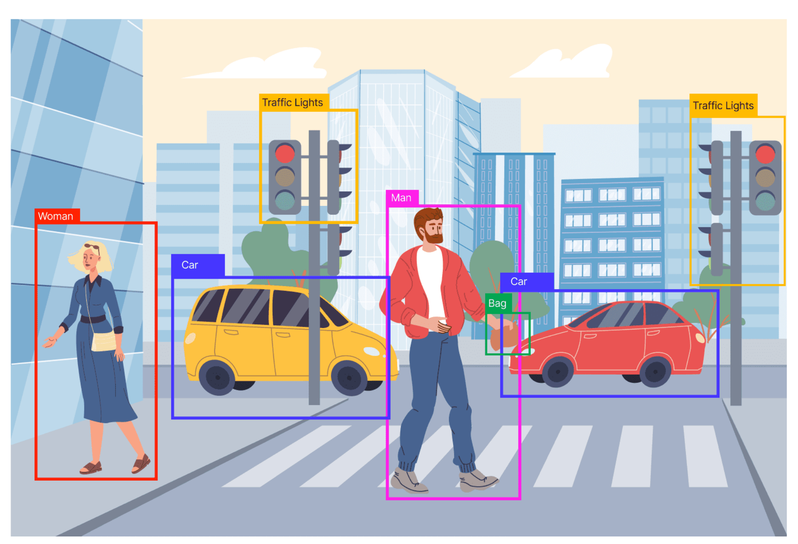
Xiaoyu Wang[1], Guojun Xiong[2], Houwei Cao[3], Jian Li[2], Yong Liu[1]
[1]New York University, [2]Stony Brook University, [3]New York Institute of Technology
{wang.xiaoyu, yongliu}@nyu.edu, {guojun.xiong, jian.li.3}@stonybrook.edu, hcao02@nyit.edu

Background & Motivation



DFL in Vehicular Networks





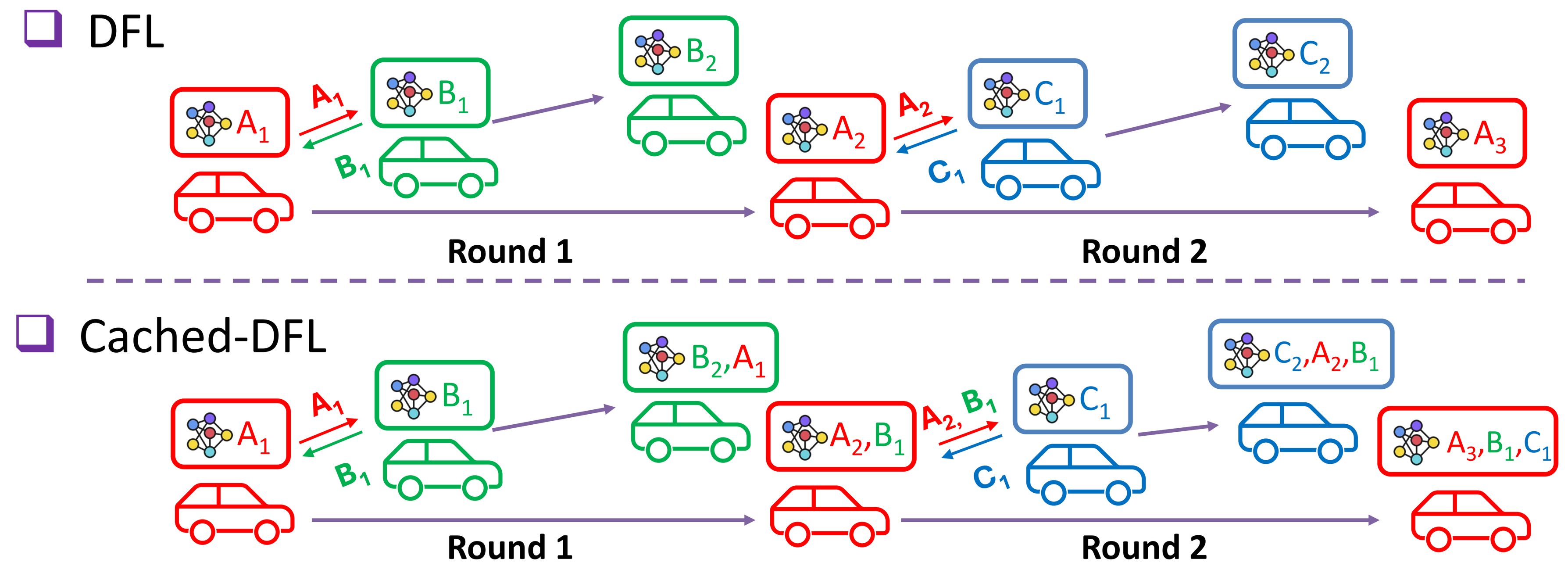
Cached-DFL

Our Proposal: Cached-DFL

- Motivated by Delay-tolerant Networking (DTN) for robust and efficient data dissemination in Mobile Ad-hoc Network (MANET);
- Knowledge Cache: stores own model + models from other agents;
- Two agents meet -> exchange/fuse local and cached models;
- Knowledge Caching-Relay: leverages mobility to accelerate model spreading/fusion globally.

Values of Caching Agent B's Model on Agent A
- ✓ Knowledge from B's unique data;
- ✓ Contribute to model fusion on A;
- ✓ **Relayed** to other agents via A;
- ✓ Fast and even model spreading for global convergence.

Costs of Model Caching
- X Communication cost
- X Storage cost
  - X Cache Replacement
- X Cached models are not update-to-date:
  - X Diverge global convergence



Caching Algorithm Design

- Data coverage vs. staleness
- *Freshness-first* Caching:
  - threshold τ_{max} -> control staleness, cached models can not be older than τ_{max}
  - when cache is full-> **LRU**
- Other Considerations:
  - data distribution/uniqueness
  - load balance
  - projected future mobility, etc.

Algorithm 2: LRU Model Cache Update (LRU Update)

Input: Current cache $\mathcal{C}_i(t)$, agent j's cache $\mathcal{C}_j(t)$, model $x_j(t)$ from agent j, current time t, cache size $\mathcal{C}_{\max}$, staleness tolerance $\tau_{\max}$

Main Process:

for each $x_k(\tau) \in \mathcal{C}_i(t)$ or $\mathcal{C}_j(t)$ **do**
  if $t - \tau \geq \tau_{\max}$ **then**
    Remove $x_k(\tau)$ from $\mathcal{C}_i(t)$ or $\mathcal{C}_j(t)$
  end if
end for
Add or replace $x_j(t)$ into $\mathcal{C}_i(t)$
for each $x_k(\tau) \in \mathcal{C}_j(t)$ **do**
  LRU Steps ...
end for
Sort models in $\mathcal{C}_i(t)$ in descending order of τ
Retain only the first $\mathcal{C}_{\max}$ models in $\mathcal{C}_i(t)$
return $\mathcal{C}_i(t+1)$

Output: $\mathcal{C}_i(t+1)$

Convergence Analysis

We also gives the convergence analysis, build a relationship related to τ_{max} : **smaller τ_{max} leads to tighter bound**

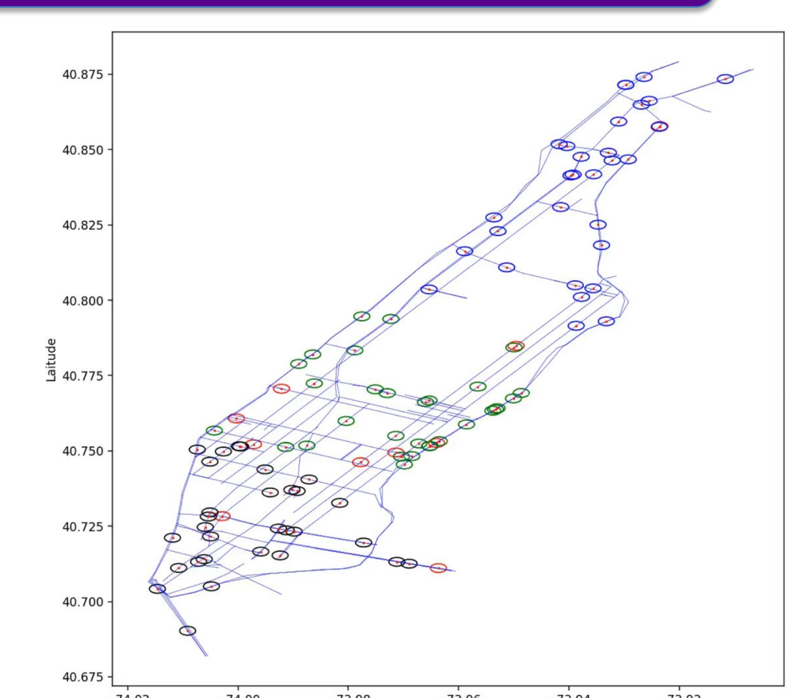
$$\min_{t=0}^{T-1} \mathbb{E}||\nabla F(x(t))||^2 \leq \frac{\tau_{max}}{\epsilon\eta C_1 KT}\mathbb{E}[F(x(0)) - F(x_{M(T)}(T)] + \mathcal{O}(\frac{\eta\rho K^2}{\epsilon C_1})$$
$$\leq \boxed{\mathcal{O}(\frac{\tau_{max}}{\epsilon\eta C_1 KT})} + \mathcal{O}(\frac{\eta\rho K^2}{\epsilon C_1}).$$

Check out more details in our **extended version**.

Evaluation

Generated Manhattan Map from **INRIX**.
100 Cars by Manhattan Mobility Model
Communication range: 100 meters
Velocity: 13.89 m/s

- NN Models: CNN
- Image Classification Datasets
  - MNIST, FashionMNIST, CIFAR-10
- Dataset Distributions:
  - i.i.d, Dirichlet, Non-i.i.d.



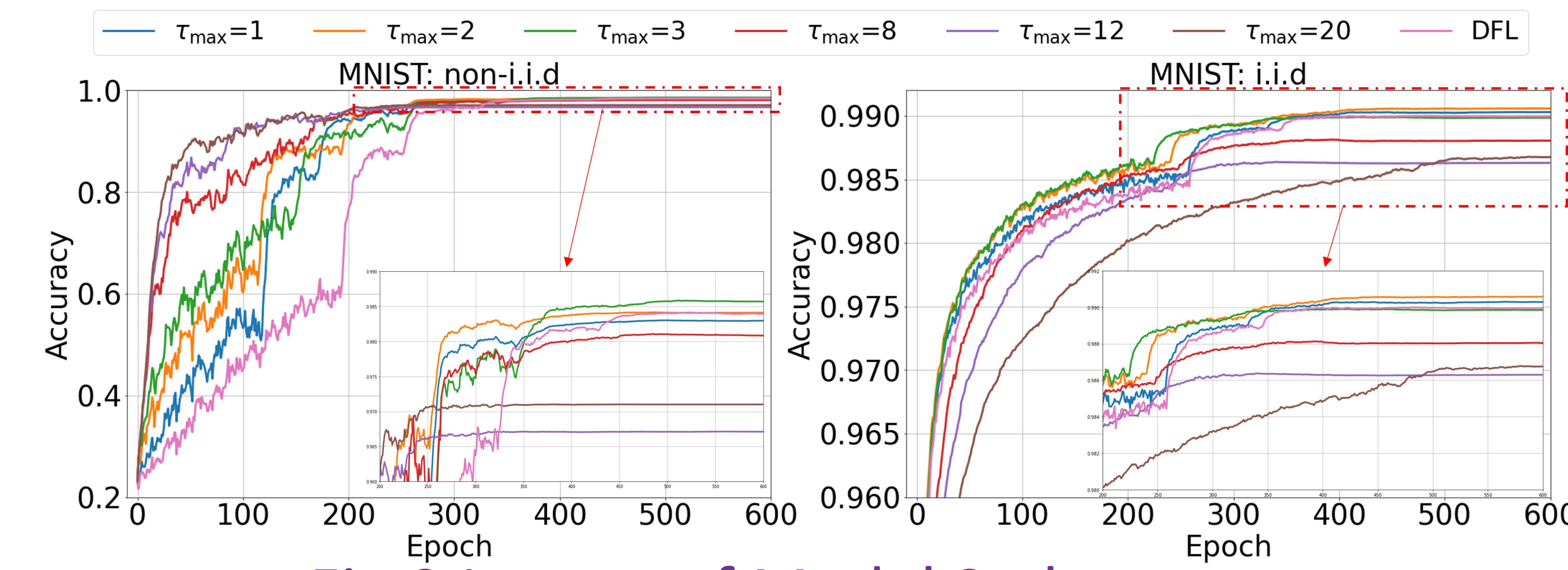


Fig.2 Impact of Model Staleness τ_{max}

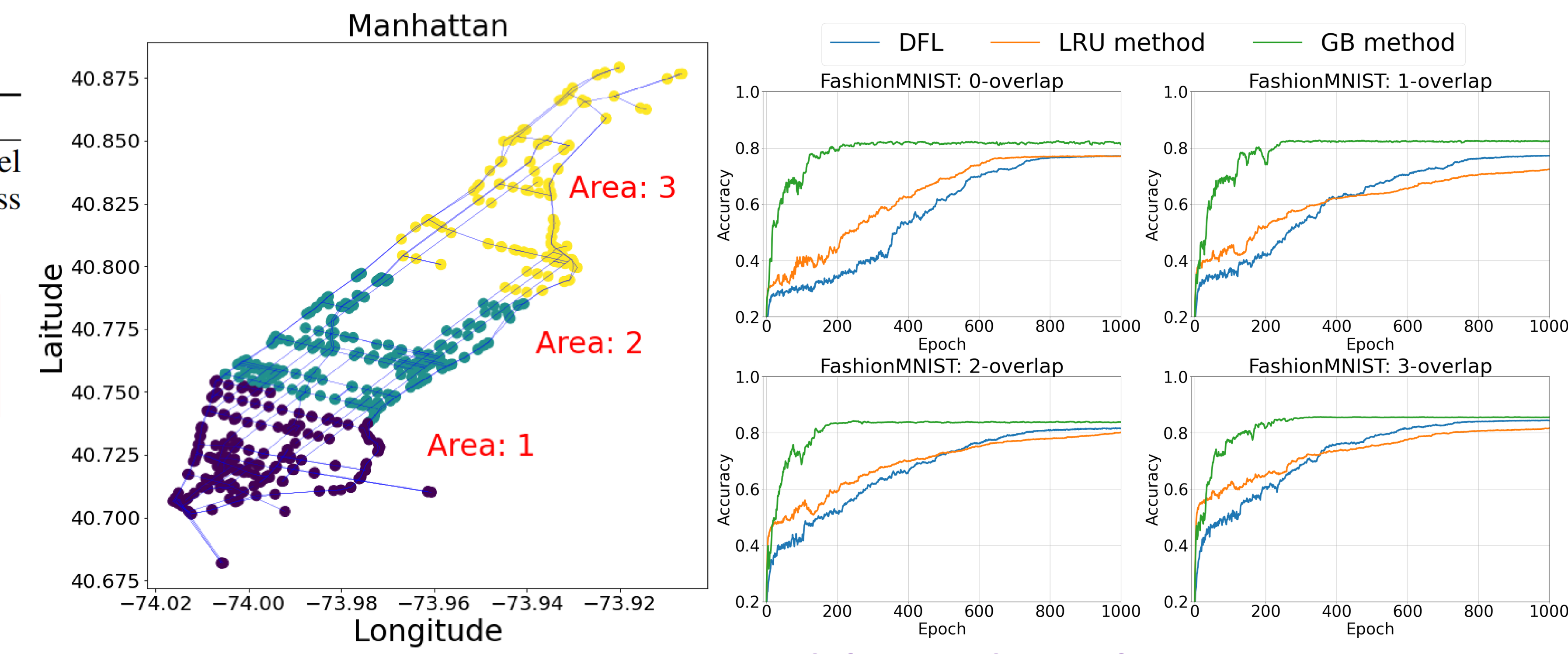


Fig.3 Grouped-based Caching

Conclusion

- Cached-DFL **outperforms** DFL w.o. caching, especially for **non-i.i.d.** data distributions on agents;
- **Larger** cache size and **smaller** model staleness τ_{max} make caching closer to the performance of CFL;
- The choice of τ_{max} should consider the **diversity data distributions** on agents;
- The **mobility** or **topology** will also have big impact on model convergence.

Extended Version: *Wang, X., Xiong, G., Cao, H., Li, J. and Liu, Y., 2024. "Decentralized Federated Learning with Model Caching on Mobile Agents", arXiv preprint arXiv:2408.14001.*

Project Code

Personal Website