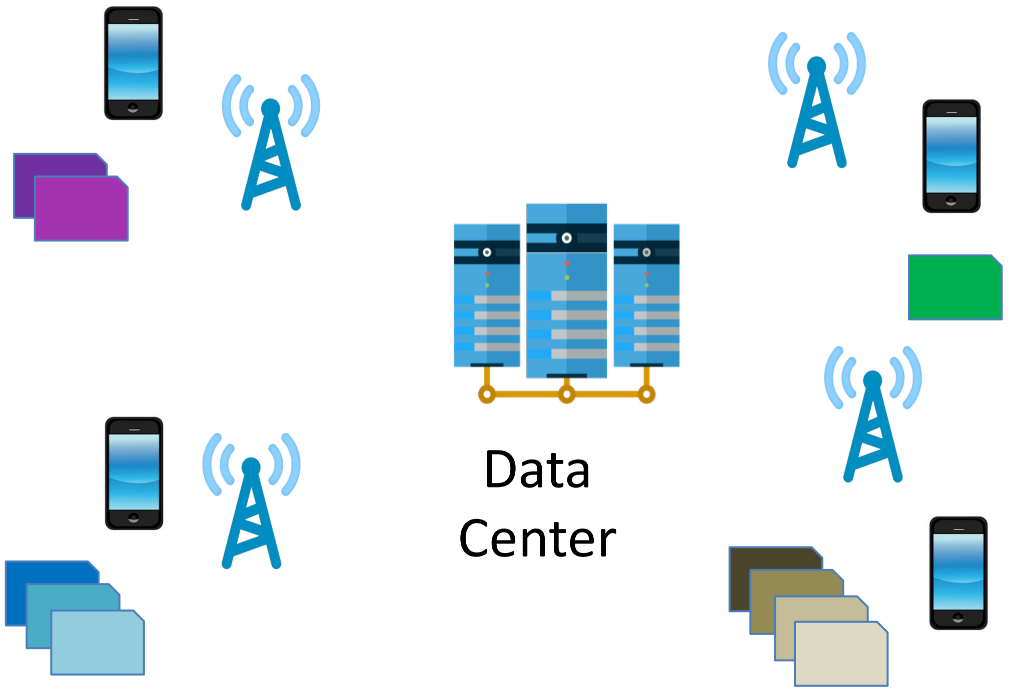
Personalized Federated Learning through Local Memorization

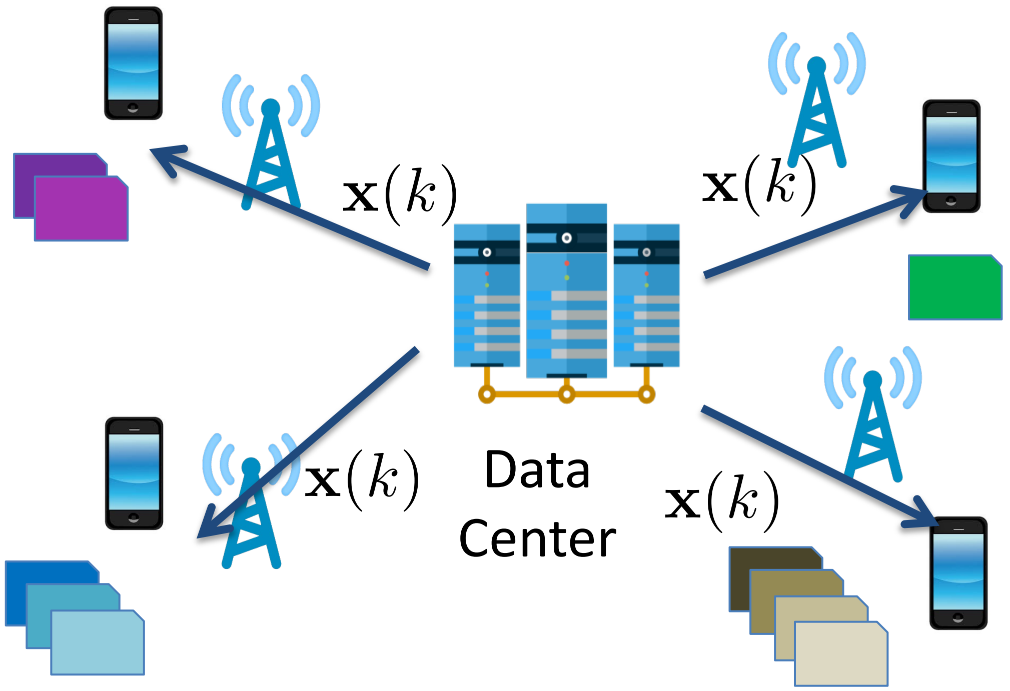
Giovanni Neglia
Joint work with O. Marfoq (Inria), L. Kameni, R. Vidal (Accenture Labs)

Classic Federated Learning



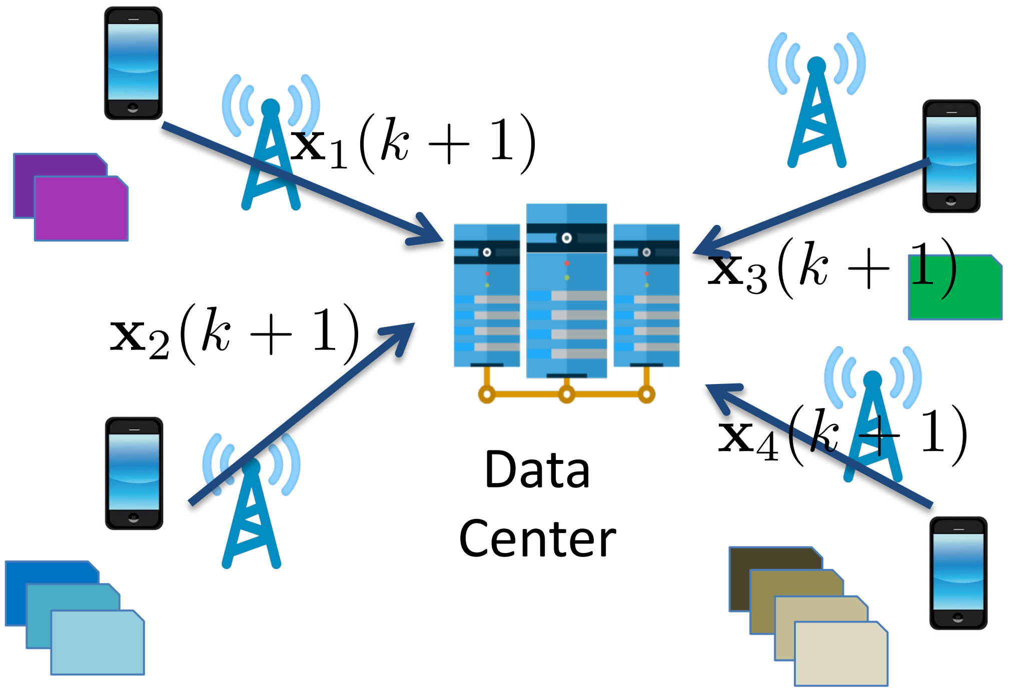
- Train ML models keeping data local
- A single model

Classic Federated Learning



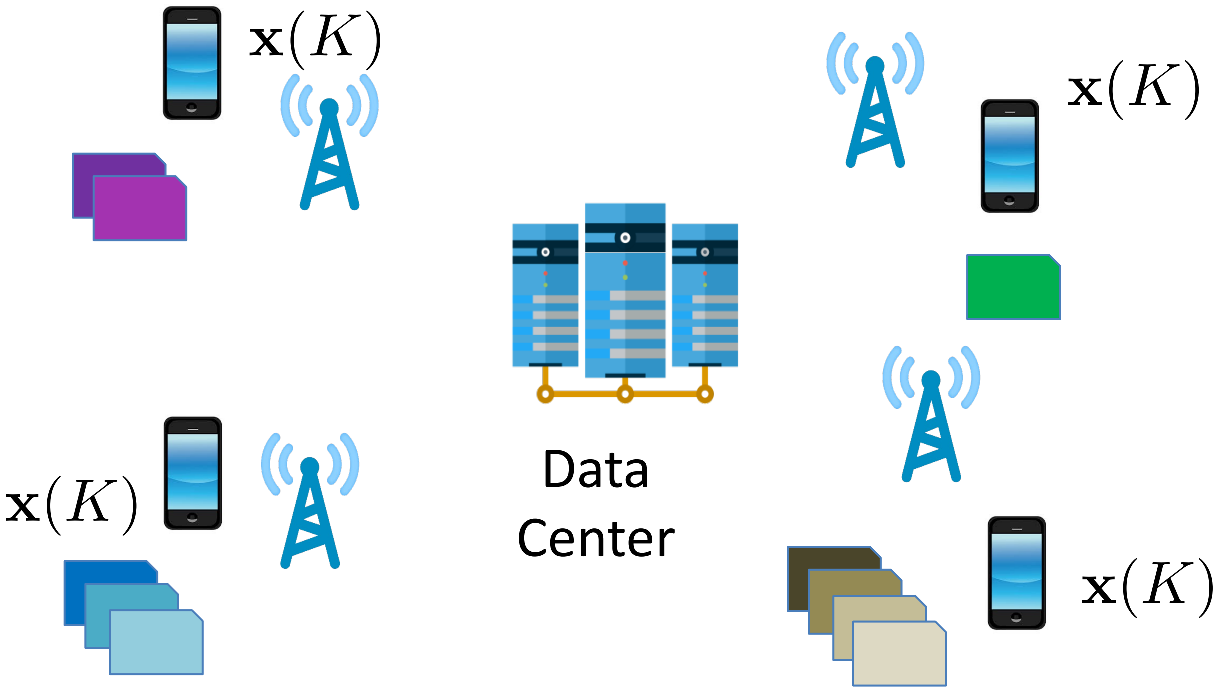
- Train ML models keeping data local
- A single model

Classic Federated Learning



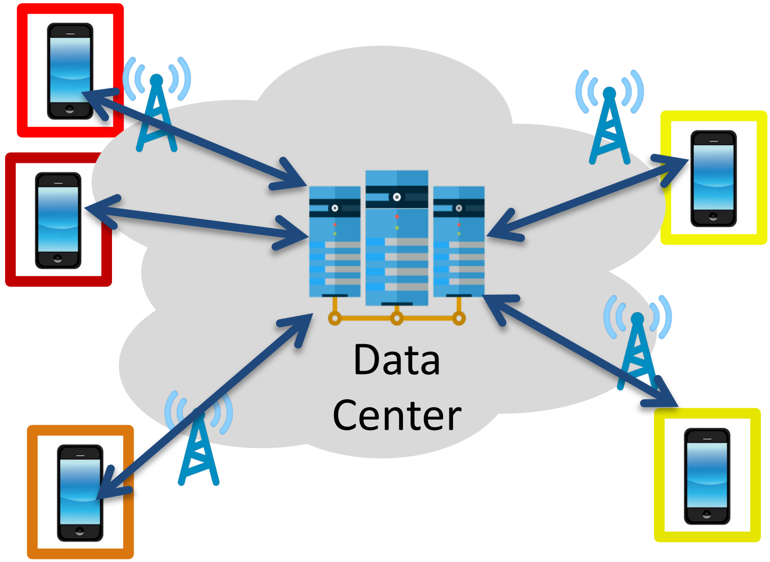
- Train ML models keeping data local
- A single model

Classic Federated Learning



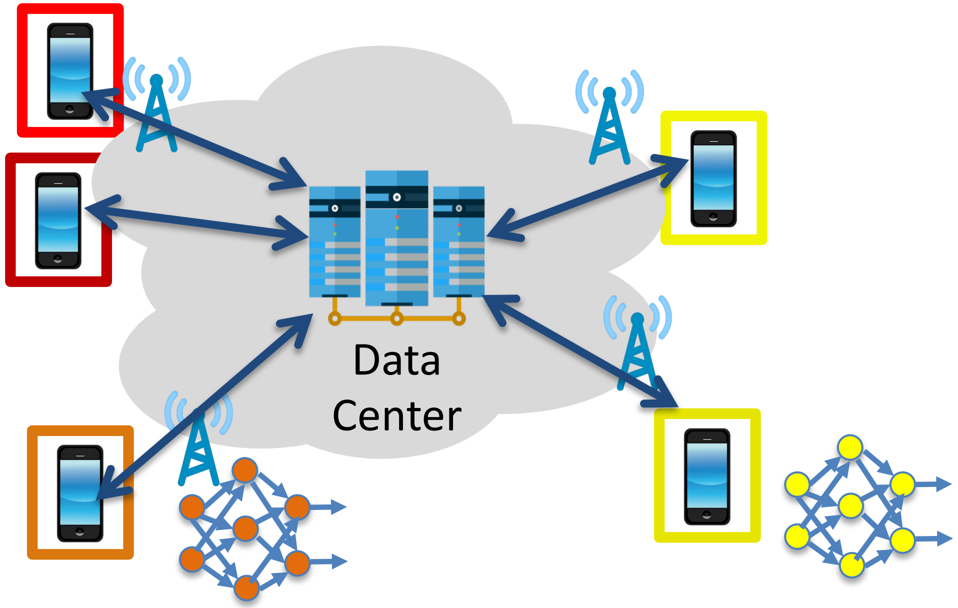
- Train ML models keeping data local
- A single model

Personalization



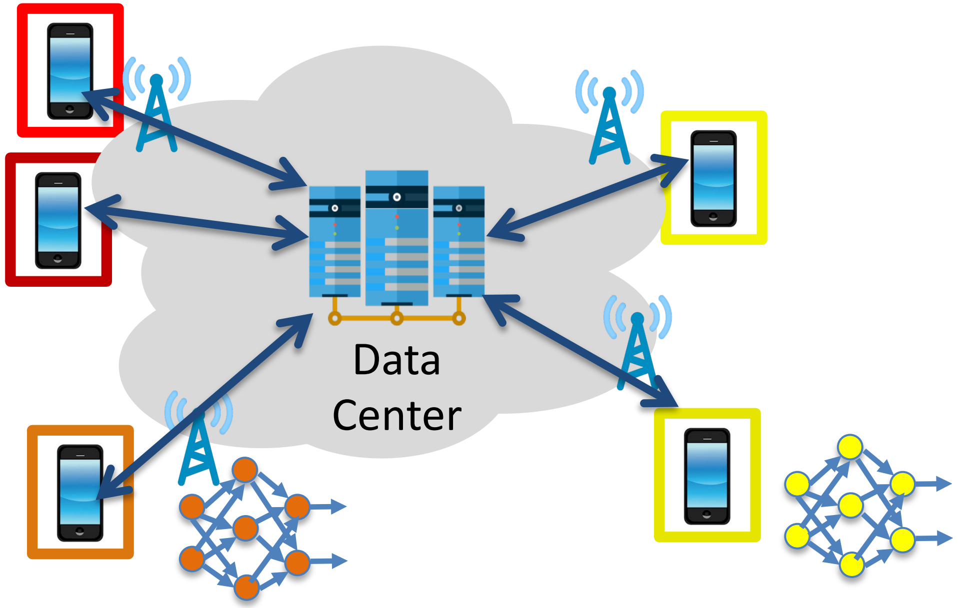
➢ Why a single model if local datasets come from different distributions? Statistical heterogeneity

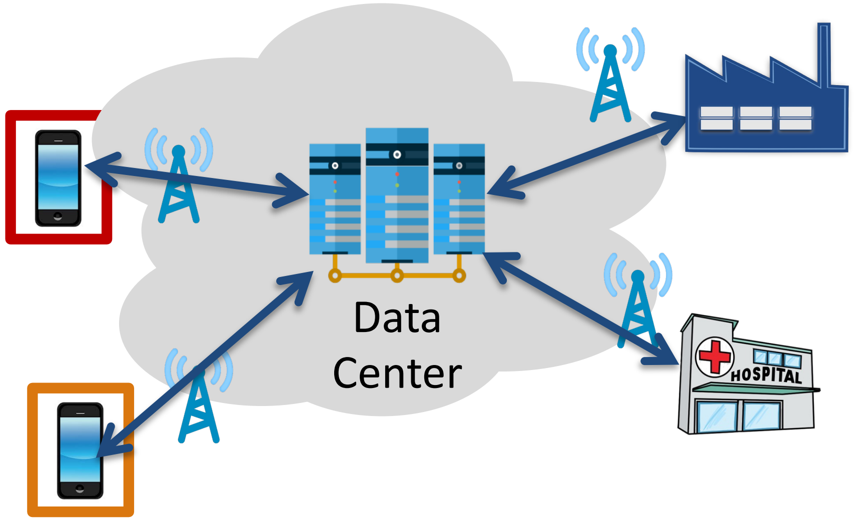
Personalization



- Why a single model if local datasets come from different distributions?
Statistical heterogeneity

Personalization

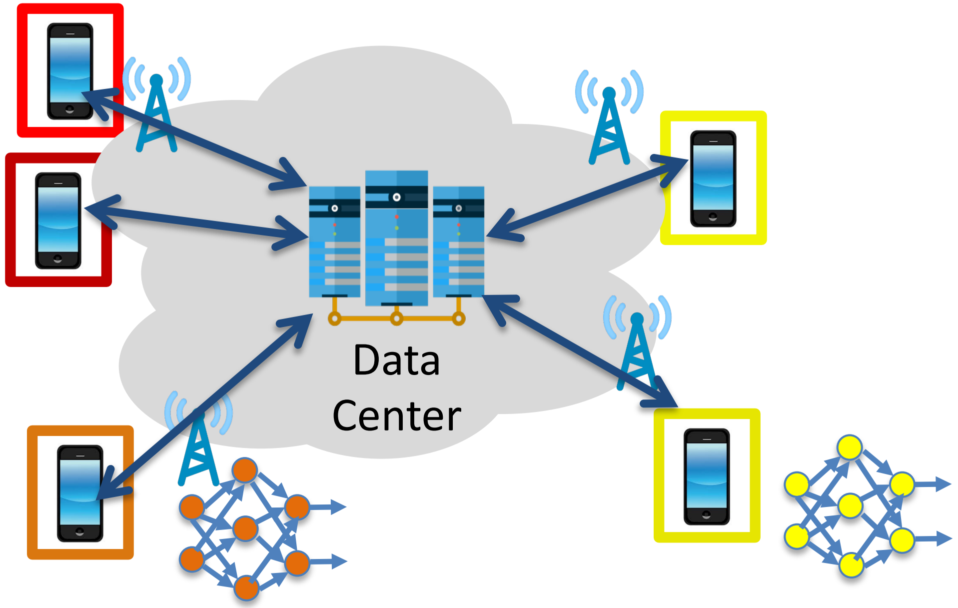




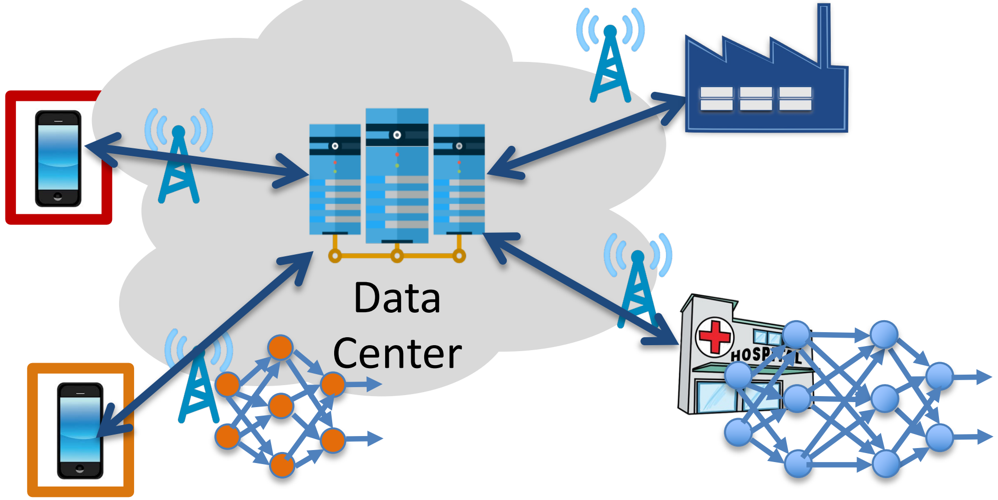
- Why a single model if local datasets come from different distributions? Statistical heterogeneity

- Why the same model architecture when clients have different capabilities? System heterogeneity

Personalization

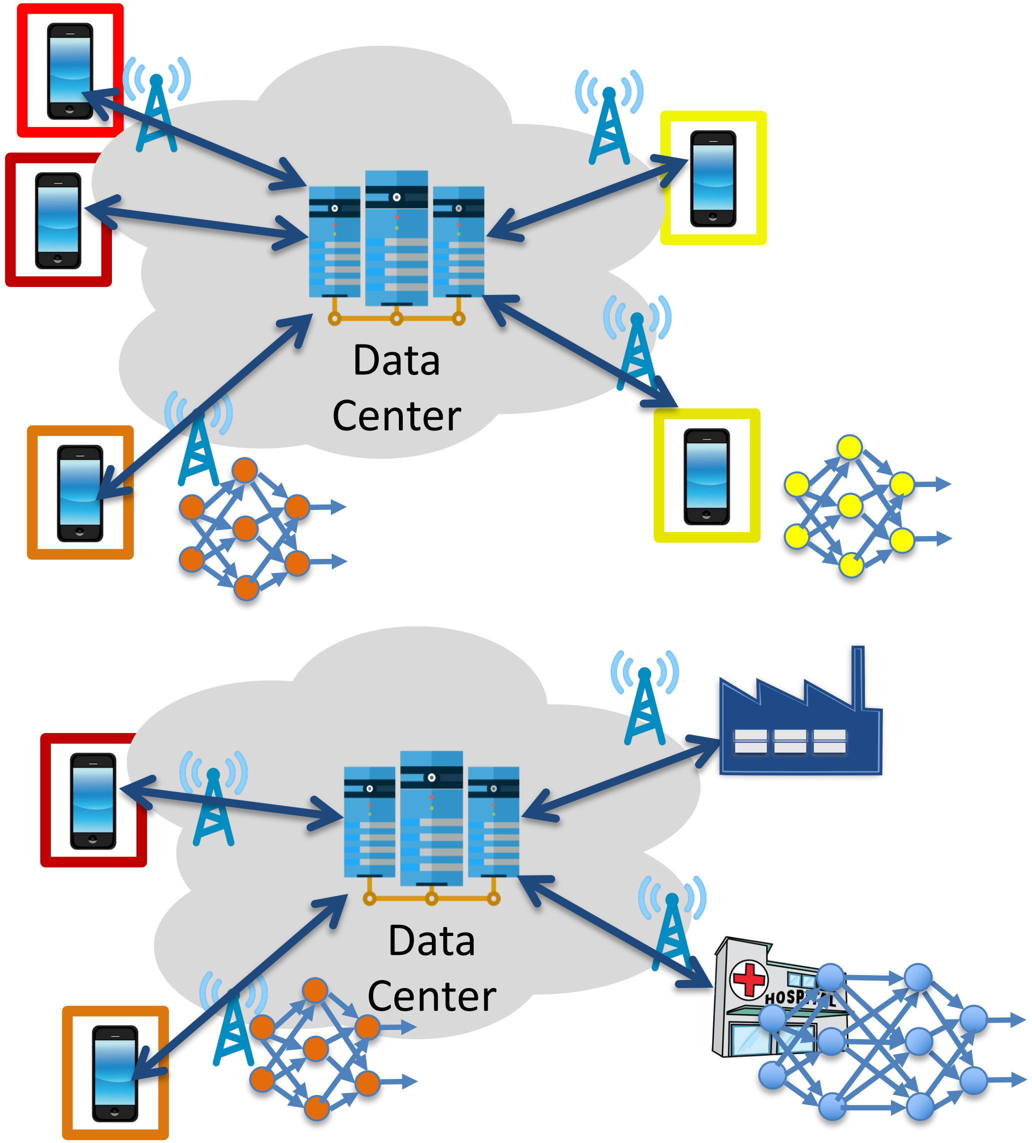


➢ Why a single model if local datasets come from different distributions? Statistical heterogeneity



➢ Why the same model architecture when clients have different capabilities? System heterogeneity

Personalization

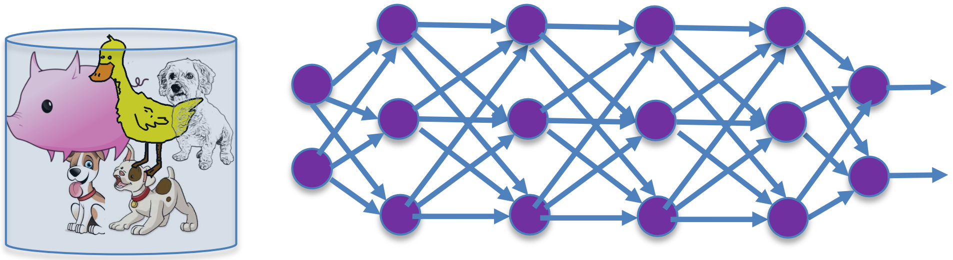


- Why a single model if local datasets come from different distributions? Statistical heterogeneity

This paper

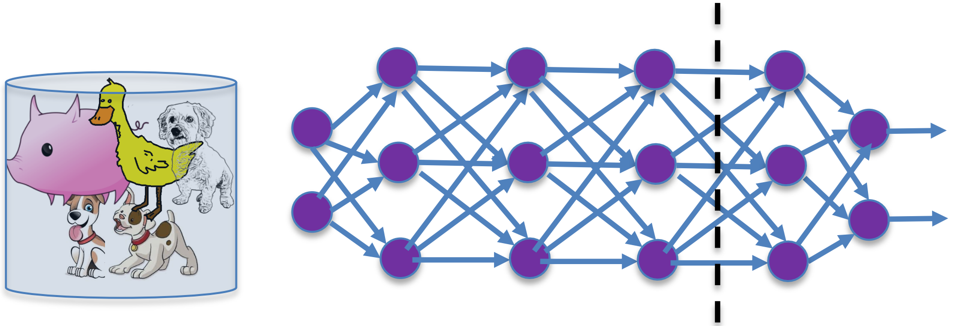
- Why the same model architecture when clients have different capabilities? System heterogeneity

Use of Memorization



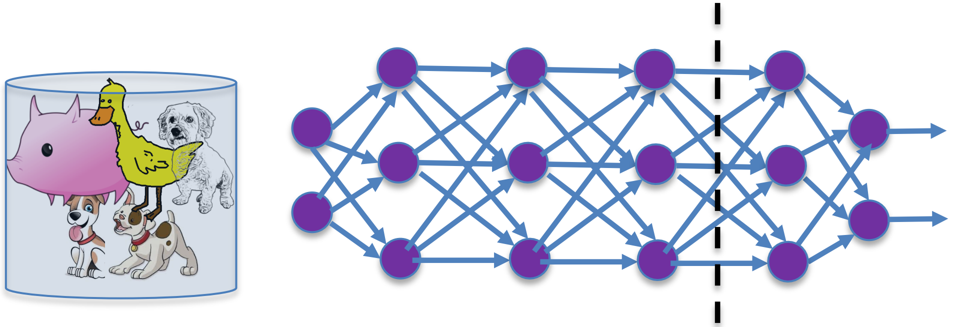
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

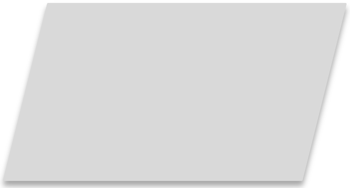
Use of Memorization



1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

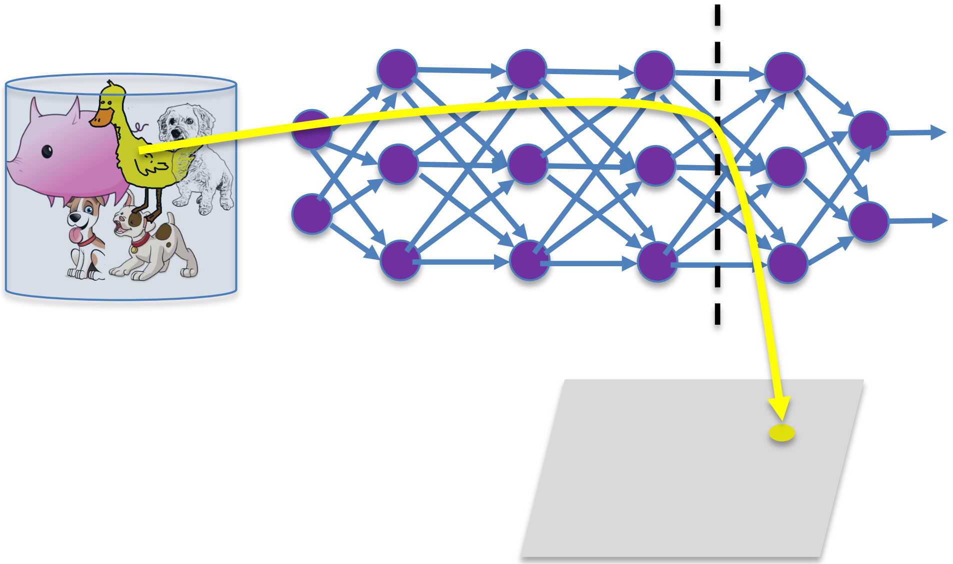
Use of Memorization





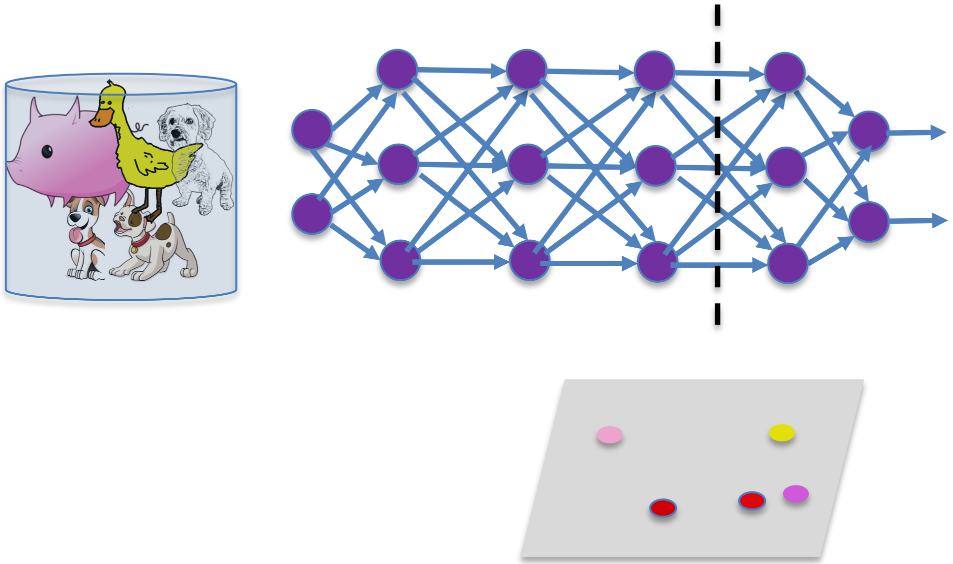
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization



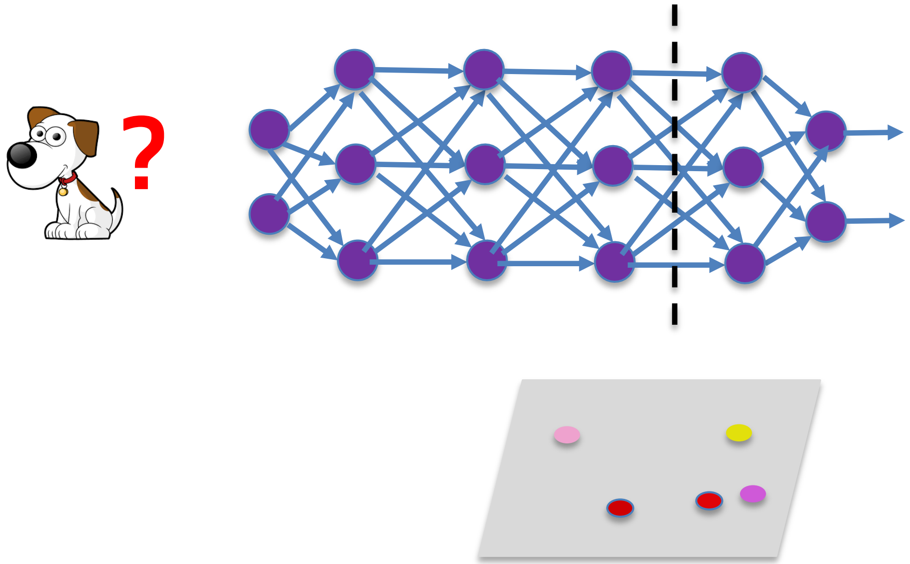
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization



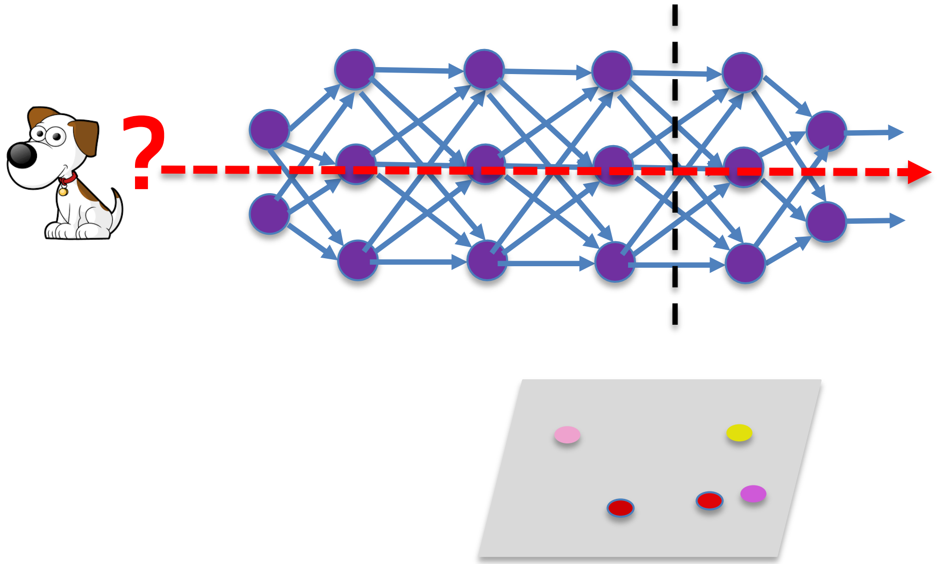
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization



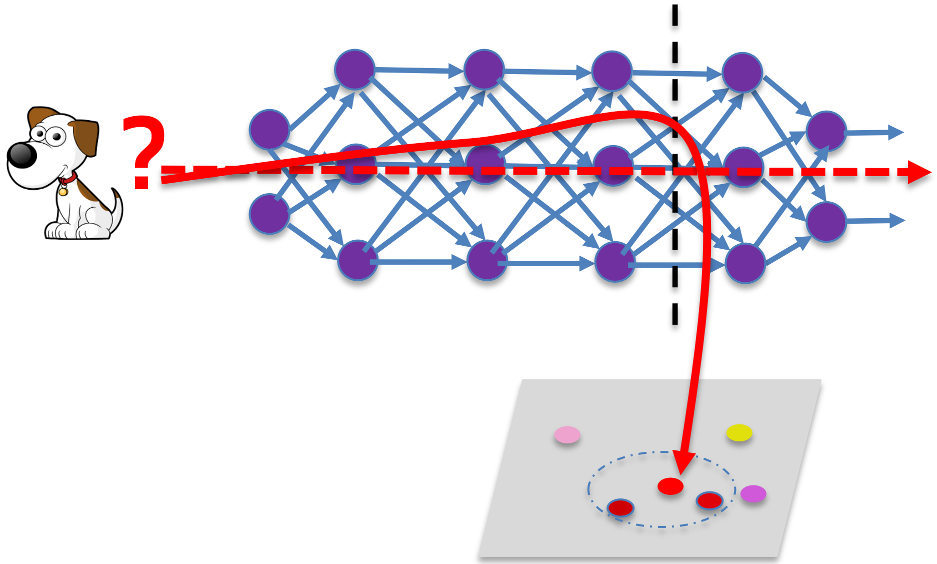
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization



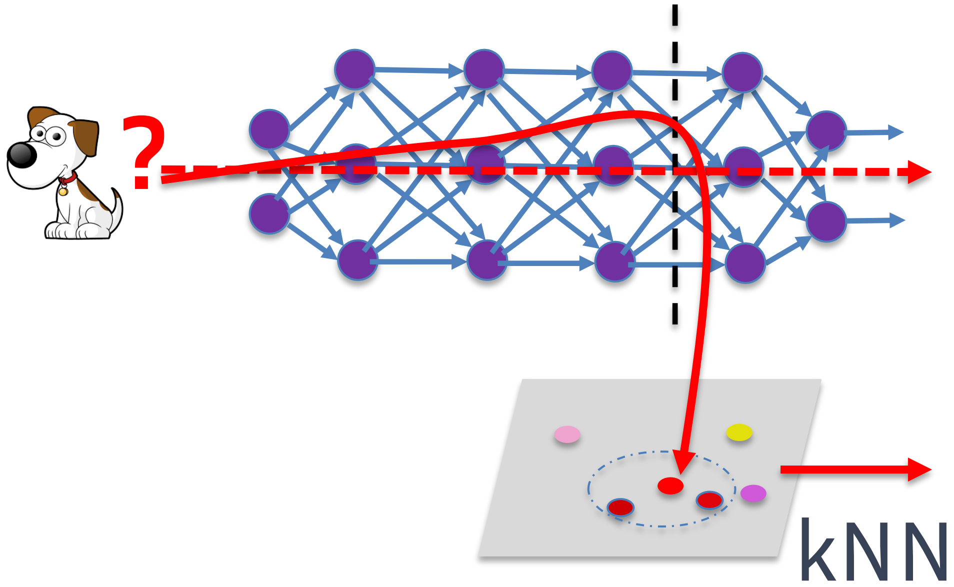
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization



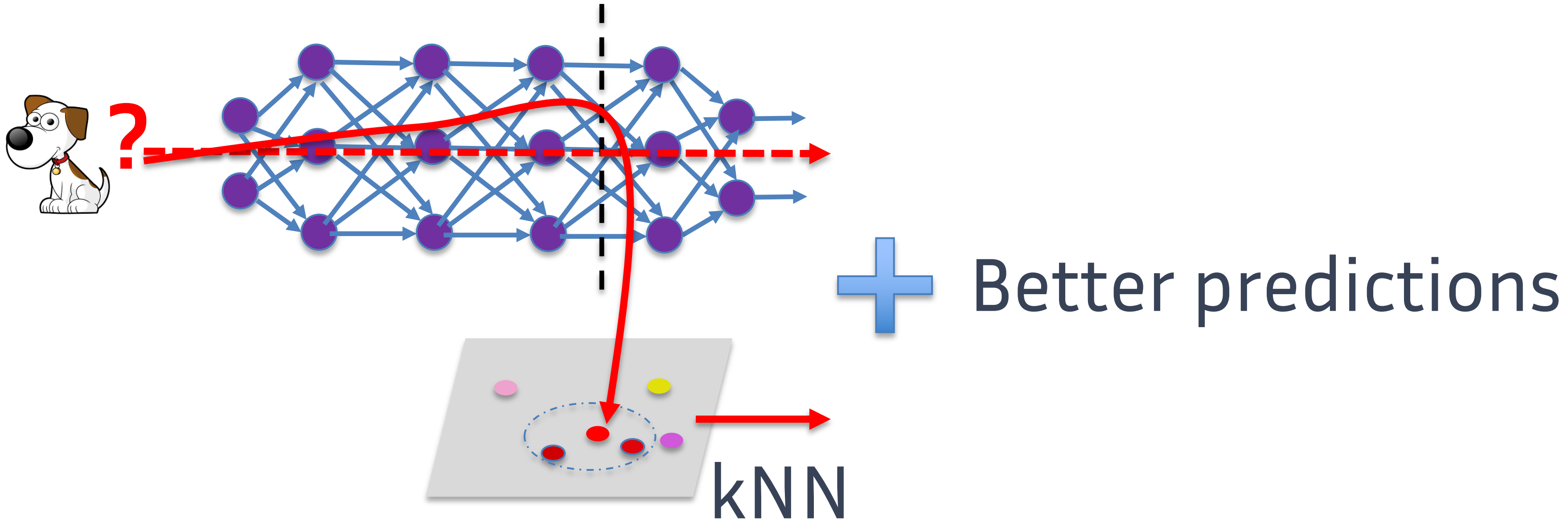
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization



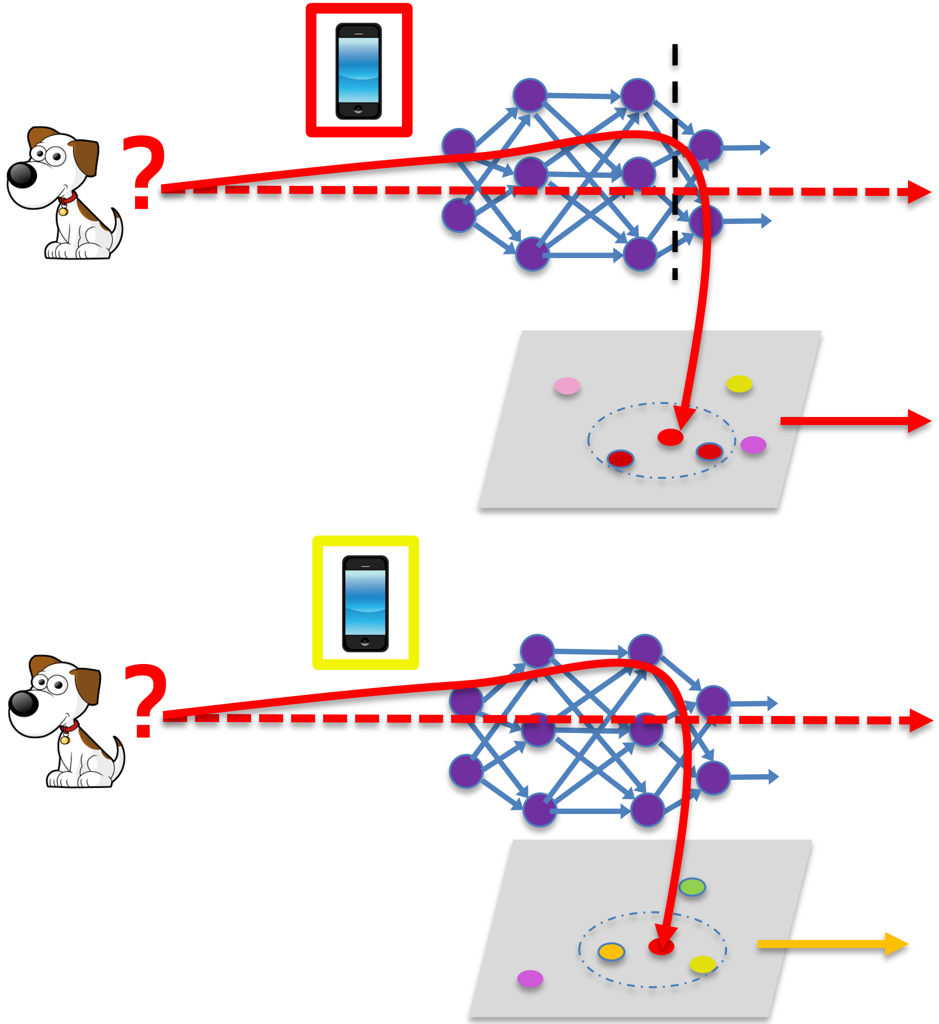
1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Use of Memorization

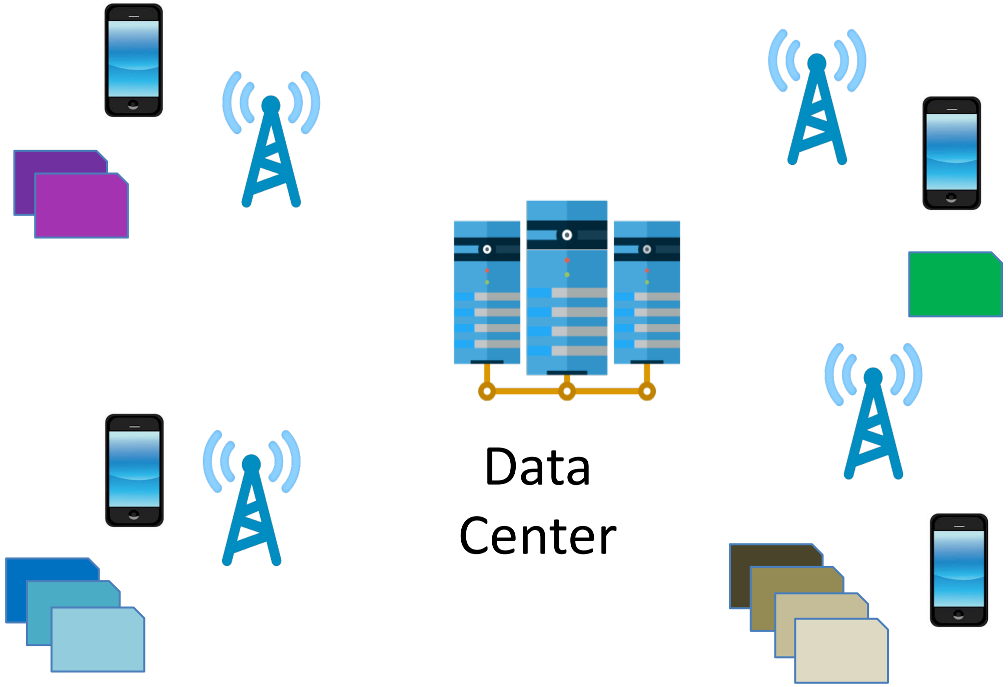


1. Khandelwal, Levy, Jurafsky, Zettlemoyer, Lewis. Generalization through Memorization: Nearest Neighbor Language Models. *ICLR'20*.
2. Orhan. A simple cache model for image recognition. *NeurIPS'18*.
3. Snell, Swersky, Zemel. Prototypical networks for few-shot learning. *NeurIPS'17*.

Our idea: Memorization for Personalization



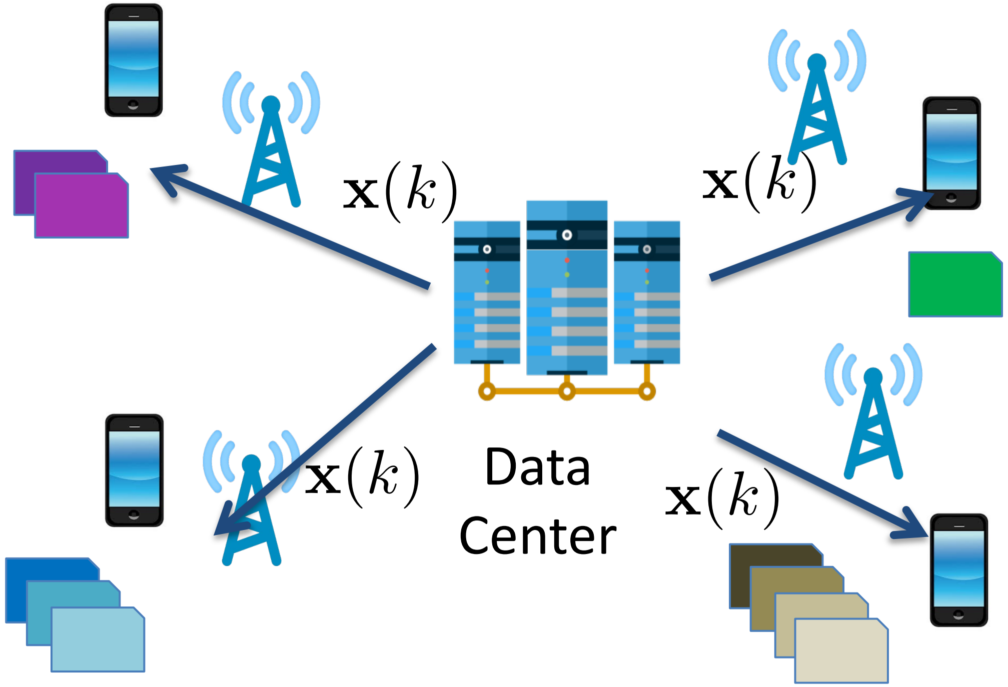
Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)

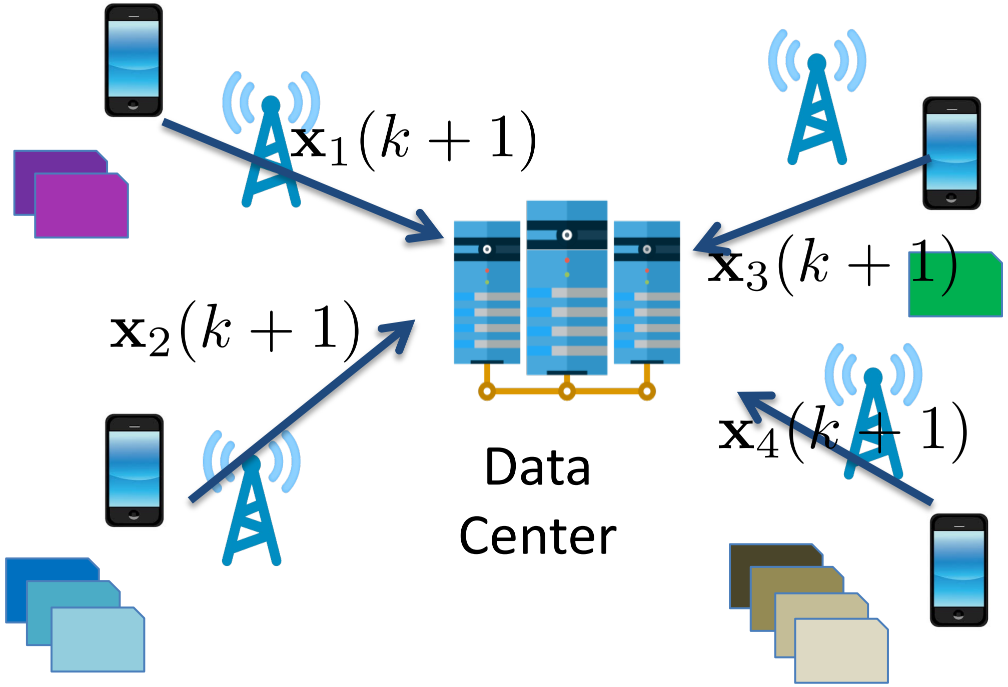
Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)

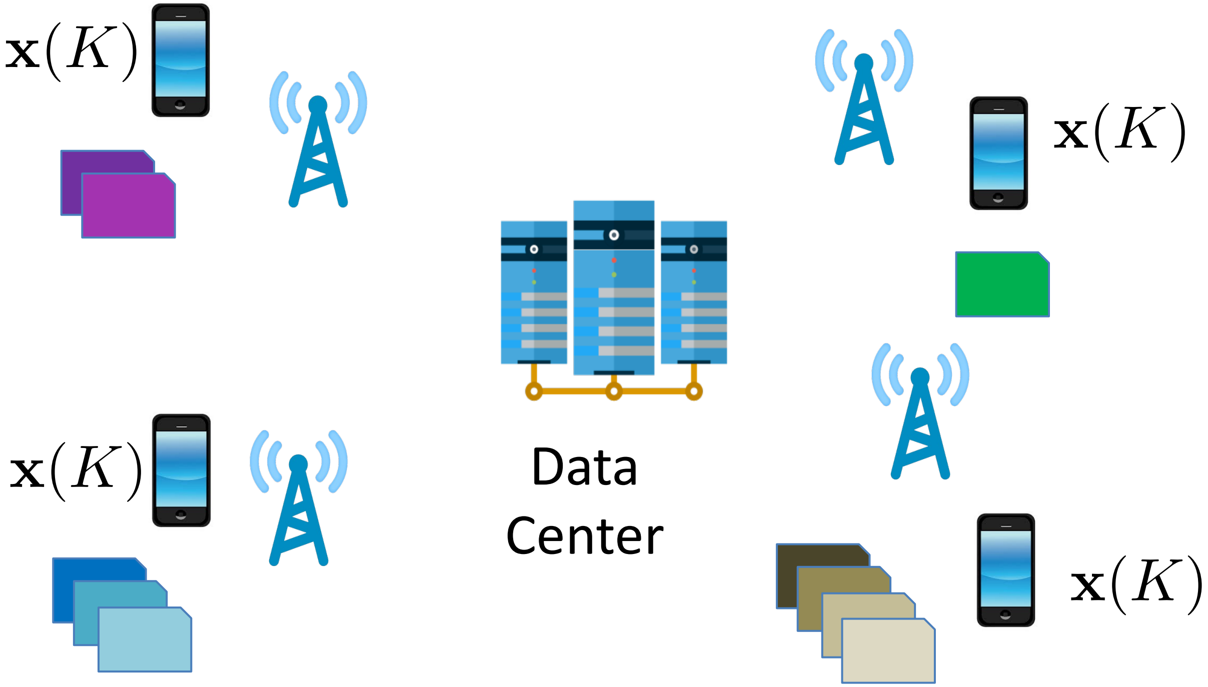
Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)

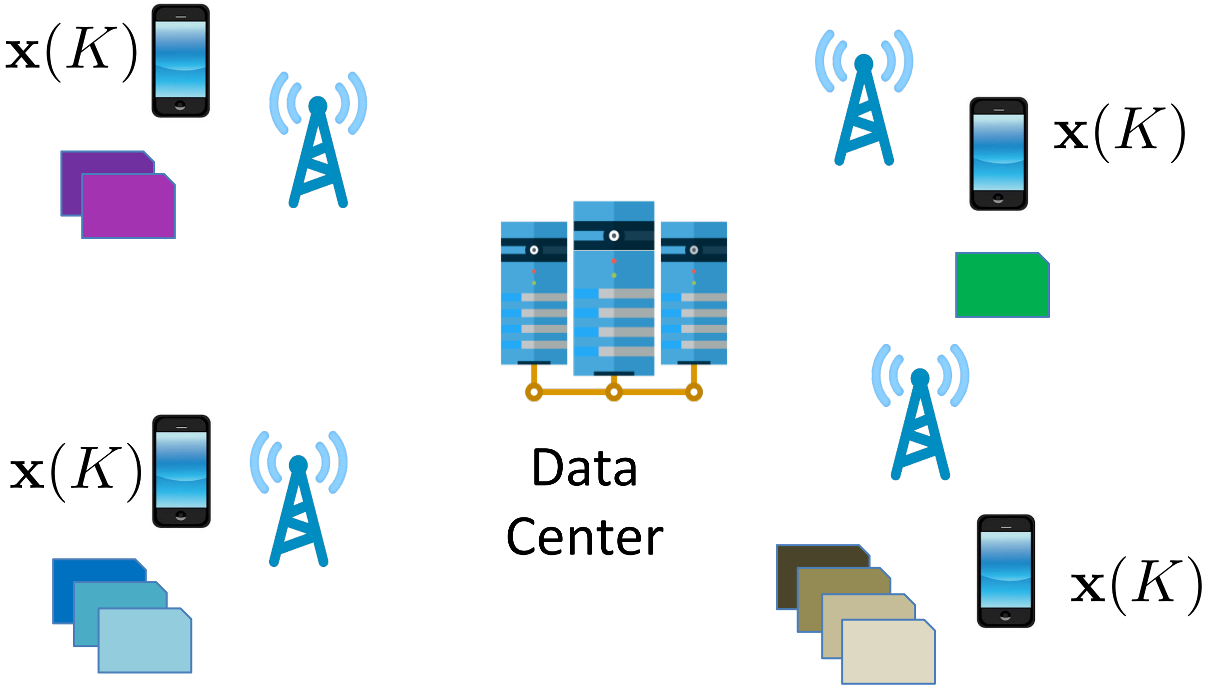
Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)

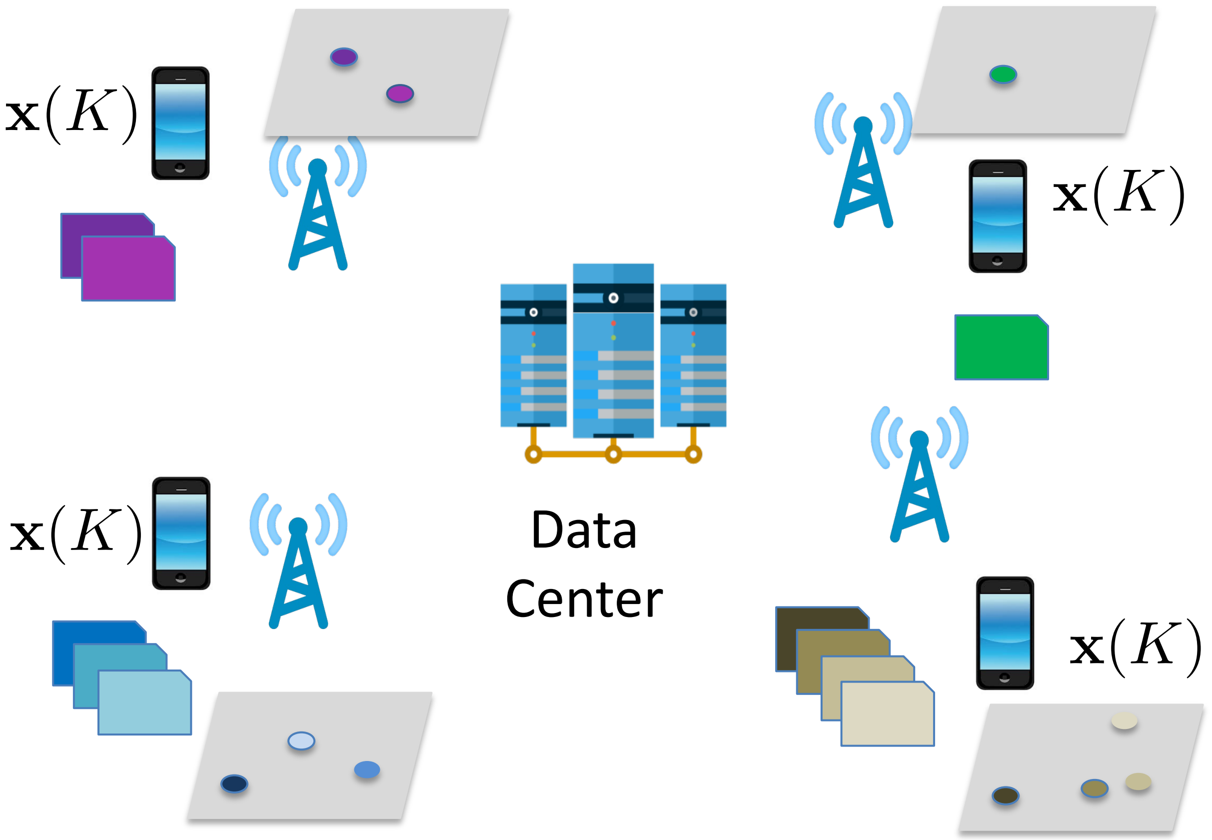
Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)
2. Each client creates its local datastore

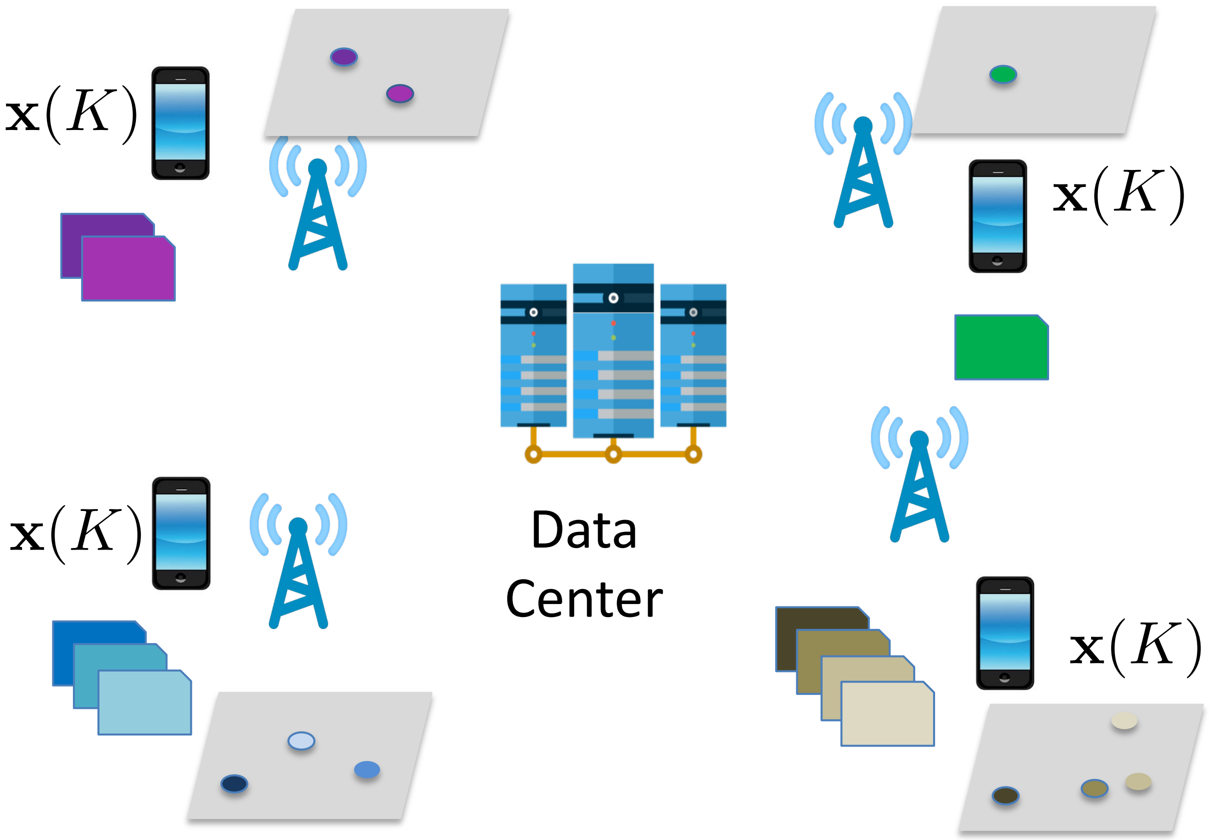
Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)
2. Each client creates its local datastore

Our Algorithm: kNN-Per



kNN-Per

1. Clients train a global model using a federated learning algorithm (e.g. FedAvg)
2. Each client creates its local datastore
3. A linear interpolation is used at inference

$$(1-\lambda)h_{\text{glob}}(\mathbf{x}(K),\chi)+\lambda h_{i,k\text{NN}}(\mathbf{x}(K),\chi)$$

Theoretical Guarantees

- Enjoys global model's convergence properties

Theoretical Guarantees

- Enjoys global model's convergence properties
- What about generalization properties?

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\otimes_{m=1}^{M}\mathcal{D}_m^{n_m}}\left[\mathcal{L}_{\mathcal{D}_m}\left(h_{m,\lambda}\right)\right] \leq (1+\lambda)\cdot\mathcal{L}_{\mathcal{D}_m}\left(h_m^*\right)$$

VC-dimension of hypothesis class

$$+\, c_1(1-\lambda)\cdot \operatorname{disc}_{\mathcal{H}}\left(\bar{\mathcal{D}},\mathcal{D}_m\right) + c_3(1-\lambda)\cdot\sqrt{\frac{d}{n}}\cdot\sqrt{c_4+\log\left(\frac{n}{d}\right)}$$

$$+\, c_2\lambda\cdot\frac{\sqrt{p}}{\sqrt[p+1]{n_m}}\cdot\operatorname{disc}_{\mathcal{H}}\left(\bar{\mathcal{D}},\mathcal{D}_m\right) + c_5\lambda\cdot\sqrt{\frac{d}{n}}\cdot\sqrt{c_4+\log\left(\frac{n}{d}\right)}\cdot\frac{\sqrt{p}}{\sqrt[p+1]{n_m}}$$

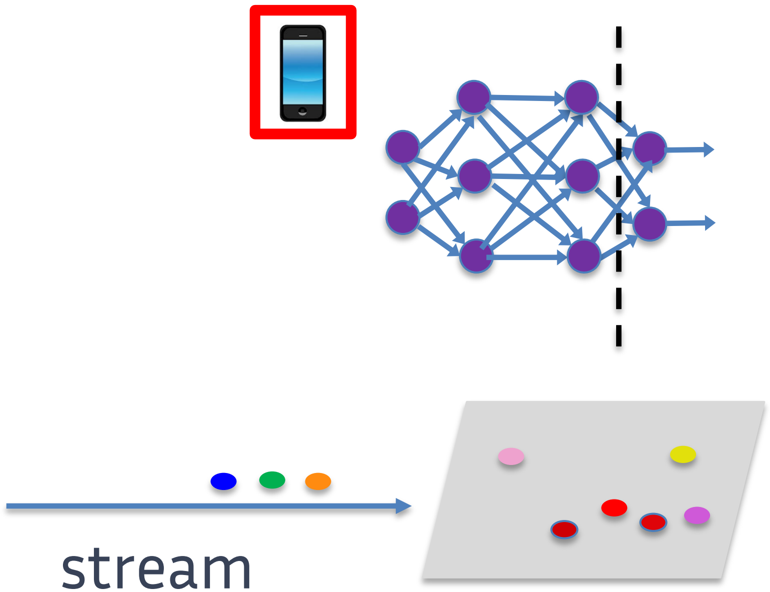
distribution heterog. aggregate dataset size local dataset size

Experiments

Table 2: Test accuracy: average across clients / bottom decile.

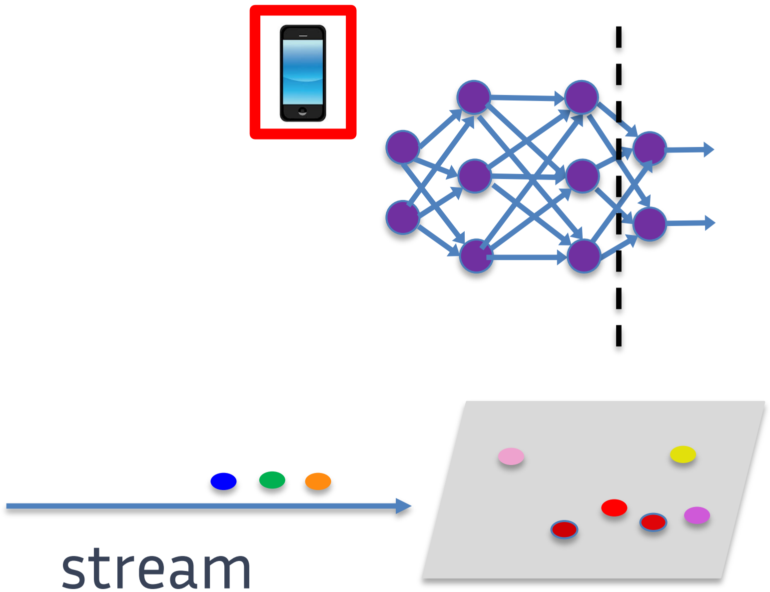
Dataset	Local	FedAvg	FedAvg+	ClusteredFL	Ditto	FedRep	APFL	kNN-Per (Ours)
FEMNIST	71.0 / 57.5	83.4 / 68.9	84.3 / 69.4	83.7 / 69.4	84.3 / 71.3	85.3 / 72.7	84.1 / 69.4	**88.2 / 78.8**
CIFAR-10	57.6 / 41.1	72.8 / 59.6	75.2 / 62.3	73.3 / 61.5	80.0 / 66.5	77.7 / 65.2	78.9 / 68.1	**83.0 / 71.4**
CIFAR-100	31.5 / 19.8	47.4 / 36.0	51.4 / 41.1	47.2 / 36.2	52.0 / 41.4	53.2 / 41.7	51.7 / 41.1	**55.0 / 43.6**
Shakespeare	32.0 / 16.0	48.1 / 43.1	47.0 / 42.2	46.7 / 41.4	47.9 / 42.6	47.2 / 42.3	45.9 / 42.4	**51.4 / 45.4**

Robustness to Distribution Shift

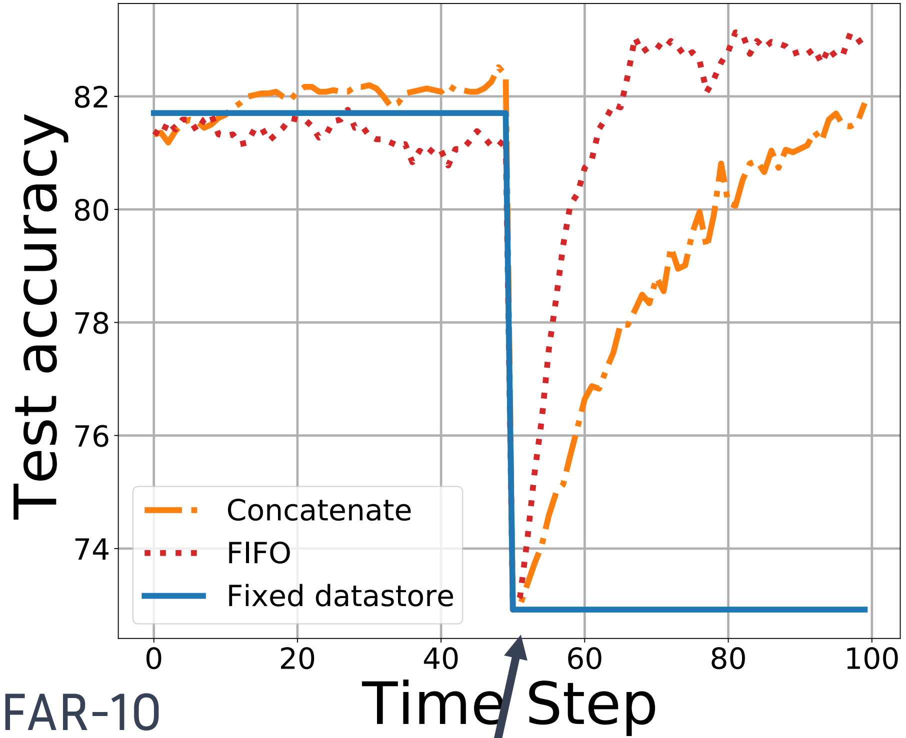


stream
of new samples

Robustness to Distribution Shift



stream
of new samples

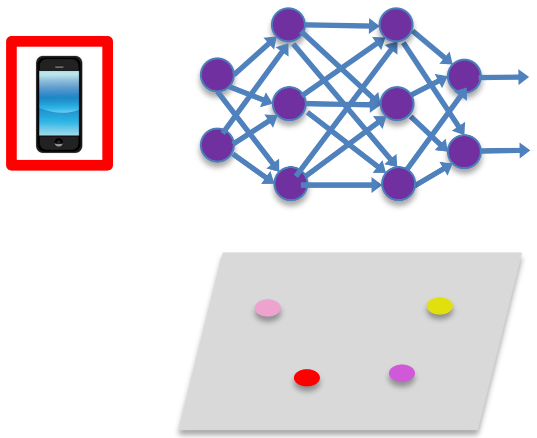


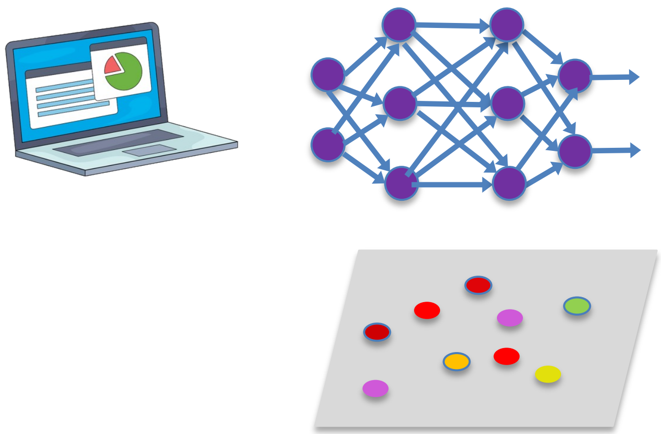
CIFAR-10

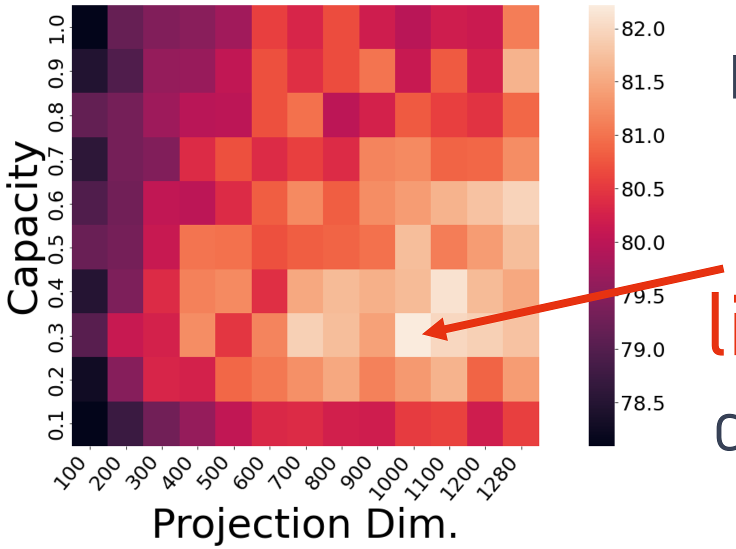
Distribution change

Datastore adapted to clients' capabilities

Datastore adapted to clients' capabilities







ProtoNN-like datastore compression

4x memory savings with limited accuracy loss (0.7pp)

CIFAR-10

Thanks for your attention

Paper

Code