Supplementary for *MAS: Towards Resource-Efficient Federated Multiple-Task Learning*

Weiming Zhuang[1]* Yonggang Wen[2] Lingjuan Lyu[1] Shuai Zhang[3]
[1]Sony AI, [2]Nanyang Technological University, [3]SenseTime Research
{weiming.zhuang,lingjuan.lv}@sony.com,ygwen@ntu.edu.sg,zhangshuai@sensetime.com

1. Experimental Details

This section provides more experimental information, including dataset, implementation details, and computation resources used.

Dataset and Federated Simulation. We run experiments using Taskonomy dataset [6], which is a large and challenging computer vision (CV) dataset of indoor scenes of buildings. To facilitate reproducibility and mitigate computational requirements, we use the tiny split of Taskonomy dataset,[1] whose size is around 445GB. We select nine CV applications to form three sets of FL tasks: `sdnkt`, `erckt`, `sdnkterca`. These nine tasks are also used in [5]. Figure 1 provides sample images of these nine FL tasks. In particular, we employ indoor images of 32 buildings [2] as the total number of clients $N = 32$; each client contains images of a building to simulate the statistical heterogeneity. Figure 2 shows sample images of four clients; their indoor scenes vary in design, layout, objects, and illumination.

Implementation Details. We reference the implementation of multi-task learning from [5]'s official repository [3] for all-in-one training and training of each split after task splitting. Each task is trained with an independent loss function. In particular, semnatic segmentation `s` uses Cross Entropy loss; surface normals and depth estimation use rotation loss based on L1 loss; keypoint detection, edge occlusion, edge texture, auto encoder, and principle curvature use L1 loss. We refer implementation of loss functions from [5] [4].

Implementation of Compared Methods. We tune the hyperparameter $\mu = 0.004$ for the proximal term in FedProx [4]. GradNorm [2] implementation is adopted from [5, 3] with default $\alpha = 1.5$ and TAG [3] implementation is adopted from their official repository [5]. Next, we provide the details of how we compute the results of HOA [5] and TAG [3].

HOA [5] needs to compute test losses for individual tasks and pair-wise task combinations for $R = 100$ rounds. After that, we use these results to estimate test losses of higher-order combinations following [5]. We then compute the actual test losses for the optimal task splits that have the lowest test losses by training them from scratch. For example, for task set `sdnkt`, we compute `s`, `d`, `n`, `k`, `t` and ten pair-wise task combinations. Then, we use these results to estimate test losses of higher-order combinations.

TAG [3] first computes all-in-one training for $R = 100$ rounds to obtain the pair-wise affinities. Then, it uses a network selection algorithm to group these FL tasks. After that, we train each split of FL tasks from scratch for $R = 100$ rounds to obtain test losses. The best result is reported for overlapping tasks. For example, {`sd, dn, kt`} is the best result of three splits of TAG on task set `sdnkt`. Then, each split is trained from scratch to obtain test losses.

Computation Resources. Experiments in this work take approximately 27,765 GPU hours of NVIDIA Tesla V100 GPU for training. We conduct three independent runs of experiments for the majority of empirical studies. In each run, task set `sdnkt` takes around 2,330 GPU hours, `erckt` takes around 3,280 GPU hours, and `sdnkterca` takes around 3,645 GPU hours. These include experiments of compared methods and ablation studies, whereas these do not include the GPU hours for validation and testing. It takes around the same GPU hours as training when we validate the model after each training round.

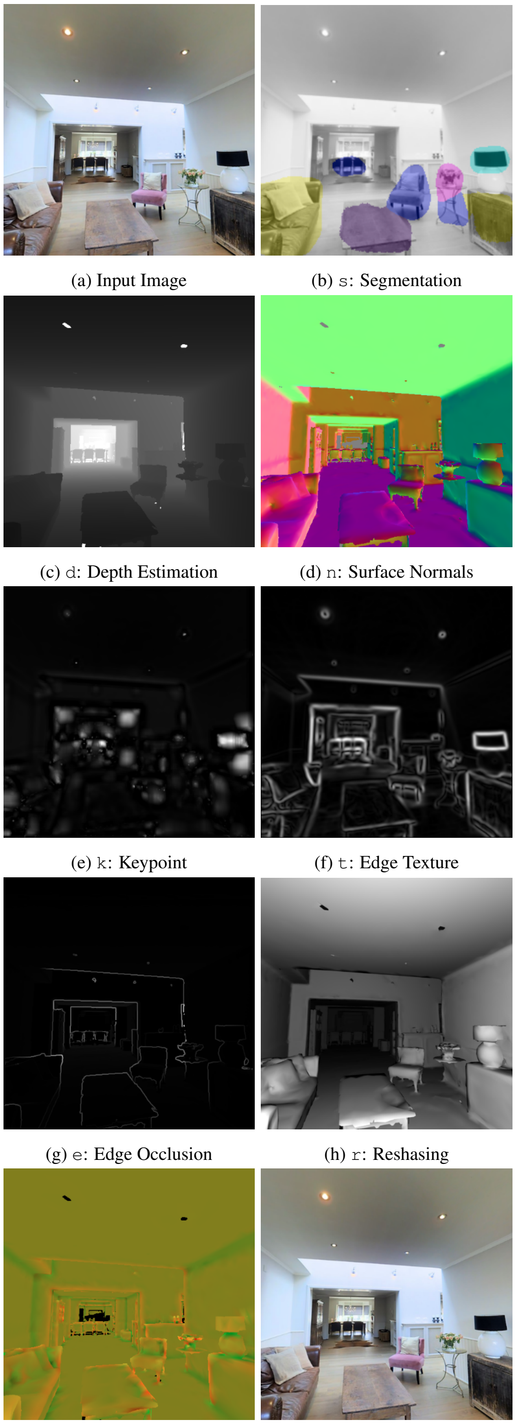
[1]Taskonomy dataset is released under MIT license and can be downloaded from their official repository https://github.com/StanfordVL/taskonomy.

[2]The name of the buildings are allensville, beechwood, benevolence, coffeen, collierville, corozal, cosmos, darden, forkland, hanson, hiteman, ihlen, klickitat, lakeville, leonardo, lindenwood, markleeville, marstons, mcdade, merom, mifflinburg, muleshoe, newfields, noxapater, onaga, pinesdale, pomaria, ranchester, shelbyville, stockman, tolstoy, and uvalda.

[3]https://github.com/tstandley/taskgrouping

[4]https://github.com/tstandley/taskgrouping/blob/master/taskonomy_losses.py

[5]https://github.com/google-research/google-research/tree/master/tag



(a) Input Image (b) s: Segmentation

(c) d: Depth Estimation (d) n: Surface Normals

(e) k: Keypoint (f) t: Edge Texture

(g) e: Edge Occlusion (h) r: Reshasing

(i) c: Principle Curvature (j) a: Auto-encoder

Figure 1: Sample images of nine FL tasks corresponding to the input image.

Figure 2: Sample images of four clients, where each client contains indoor scenes of a building. These indoor images differ in design, layout, objects, and illumination.

2. Additional Experimental Evaluation

This section provides more experimental results, including comprehensive results of performance evaluation and additional ablation studies.

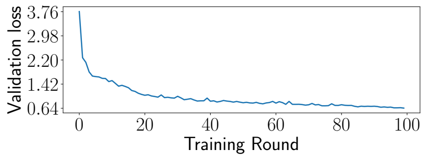
2.1. Performance Evaluation

Table 3, 4, and 5 provide comprehensive comparison of different methods on test loss, training time, and energy consumption on task sets `sdnkt`, `erckt`, and `sdnkterca`, respectively. They complement the results in the main manuscript. Additionally, these tables also provide carbon footprints (CO2eq) of different methods. The carbon footprints are estimated using Carbontracker [1].[6] Our method reduces around 40% on carbon footprints on these three task sets compared with one-by-one training; it reduces 1526g CO_2eq or equivalent to traveling 12.68km by car on `sdnkterca`. The reduction is even more significant when compared with TAG and HOA. Although we run experiments using Tesla V100 GPU, the relative results of energy and carbon footprint among different methods should be representative of the scenarios of edge devices.

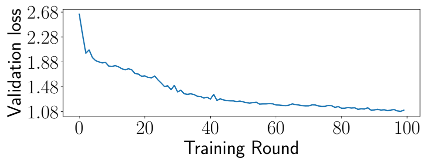
2.2. Additional Analysis and Ablation Studies

This section presents additional analysis of MAS and provides additional ablation studies.

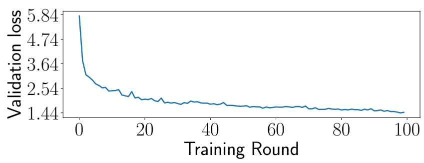
[6] Carbon intensity of a training varies over geographical regions according to [1]. We use the national level (the United Kingdom as the default setting of the tool) of carbon intensity for a fair comparison across different methods. These carbon footprints serve as a proxy for evaluation of the actual carbon emissions.



(a) Task set `sdnkt`



(b) Task set `erckt`



(c) Task set `sdnkterca`

Figure 3: Changes of validation loss over the course of training on task sets: (a) `sdnkt`, (b) `erckt`, and (c) `sdnkterca`. Validation loss converges as training proceeds.

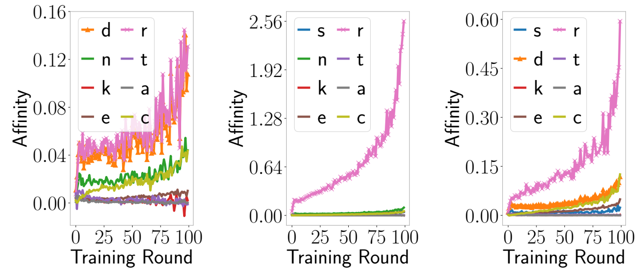
Method	Task Set	Two Splits	Three Splits	Four Splits	Five Splits
TAG	`sdnkt`	sdn,kt	sd,dn,kt	sd,sdn,dn,kt	
MAS	`sdnkt`	sdn,kt	sdn,k,t	sd,n,k,t	s,d,n,k,t
TAG	`erckt`	er,rckt	er,kt,rc	er,kt,rc,rt	
MAS	`erckt`	er,ckt	er,c,kt	er,c,k,t	e,r,c,k,t
TAG	`sdnkterca`	sdnkterca,dr	sdnerc,dr,kta	sc,dr,ne,kta	
MAS	`sdnkterca`	snkteac,dr	snec,dr,kta	sn,dr,ka,etc	sn,dr,ka,e,tc

Table 1: Task splitting results of TAG [3] and MAS on task sets `sdnkt`, `erckt`, and `sdnkterca`. Each split is separated by a comma.

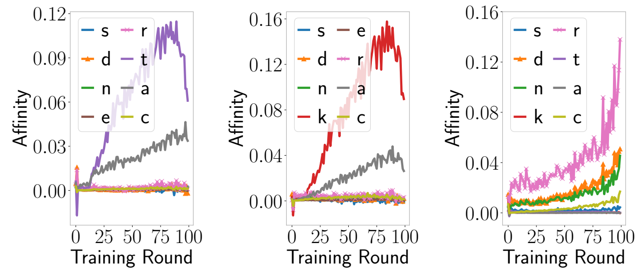
Task Set	Splits	Optimal Splits			Worst Splits		
`sdnkt`	2	dk,snt	sn,dkt	nt,sdk	st,dnk	st,dnk	st,dnk
	3	t,sn,dk	k,t,sdn	d,sn,kt	d,st,nk	d,st,nk	s,dt,nk
`erckt`	2	r,eckt	t,erck	et,rck	rk,ect	ek,rct	e,rckt
	3	r,ec,kt	r,t,eck	r,ec,kt	c,e,rk	e,k,rct	e,rt,ck

Table 2: Results of the optimal and worst splits in three runs of experiments. They are not identical due to variances in three runs of experiments.

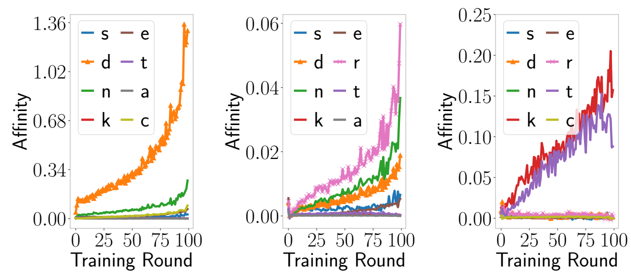
Changes of Vadiation Loss. Figure 3 presents validation losses over the course of all-in-one training of three FL task sets `sdnkt`, `erckt`, and `sdnkterca`. It shows that validation losses converge as training proceeds.



(a) Affinities to `s` (b) Affinities to `d` (c) Affinities to `n`



(d) Affinities to `k` (e) Affinities to `t` (f) Affinities to `e`

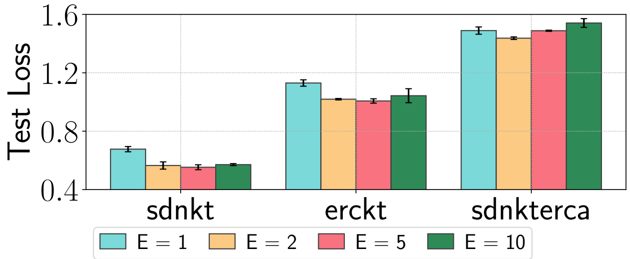


(g) Affinities to `r` (h) Affinities to `c` (i) Affinities to `a`

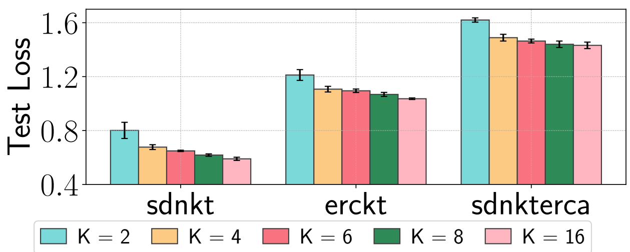
Figure 4: Changes of affinity scores of one FL task to the other over the course of training on task set `sdnkterca`. The trends of affinities emerge at the early stage of training.

Splitting Results of Various Methods. We provide results of task splitting of TAG [3] and MAS in Table 1. Table 2 presents the splitting results of the optimal and worst splits. They are not identical due to variances in multiple runs of experiments. We report the mean and standard deviation of test losses of the optimal splits and the worst splits in the manuscript. The large variances of the optimal and worst splits suggest the instability of splitting by measuring the performances of training from scratch in the FL settings and demonstrate the advantage of our methods in obtaining stable splits.

Dataset Size and Performance. The dataset size of task set `sdnkt` is around 315GB in our experiments, compared to 2.4TB of dataset used in experiments of TAG [3]. The test loss of ours (0.512 in Table 2 in the main manuscript), however, is better than the optimal one in TAG paper [3] (0.5246). This back-of-the-envelope comparison indicates



(a) Impact of E



(b) Impact of K

Figure 5: Analysis of the impact of local epoch E and impact of the number of selected clients K. Larger E (with fixed $R = 100$) and K requires higher computation. They could reduce losses, but the marginal benefit decreases as computation increases.

the potential to extend our approaches to multi-task learning. Besides, it could also suggest that our data size is sufficient for evaluation.

Impact of Affinity Computation Frequency ρ. The frequency of computing affinities in Equation 3 determines the amount of extra needed computation. We use $\rho = 5$ and compute affinities for the first ten rounds for all experiments because the trend of affinities emerges in the early stage of training in Figure 4. It would increase the computation of all-in-one training by around 2%, which is already factored into the energy consumption computation in previous experiments. The results in Table 3, 4, and 5 show that MAS is effective with this setting and the amount of computation is acceptable.

Impact of Local Epoch. Figure 5a show the impact of local epoch E on task sets `sdnkt`, `erckt`, and `sdnkterca`. They complement results of task set `sdnkt` in the main manuscript. Larger E could lead to better performance with fixed $R = 100$. It is especially effective when increasing $E = 1$ to $E = 2$, but further increasing E could degrade the performance. It indicates that simply increasing computation has limited capability to improve performance.

Impact of The Number of Selected Clients. Figure 5b compares the performance of different numbers of selected clients $K = \{2, 4, 6, 8, 16\}$ on three task sets `sdnkt`, `erckt`, and `sdnkterca`. The results on three FL task sets are similar; increasing K reduces losses, but the marginal benefit decreases as K increases.

References

[1] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, July 2020. arXiv:2007.03051. 2

[2] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018. 1

[3] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

[4] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 1

[5] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020. 1

[6] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 1

Method	Total Loss	Time (h)	Energy (kWh)	CO2eq (g)	s	d	n	k	t
One-by-one	0.603 ± 0.030	16.9 ± 0.5	8.4 ± 0.1	2465 ± 39	0.086 ± 0.005	0.261 ± 0.023	0.107 ± 0.001	0.107 ± 0.003	0.043 ± 0.002
FedAvg*	0.677 ± 0.018	7.3 ± 0.3	3.7 ± 0.1	1086 ± 28	0.087 ± 0.002	0.246 ± 0.010	0.136 ± 0.001	0.126 ± 0.019	0.083 ± 0.008
FedProx*	0.711 ± 0.070	7.7± 0.5	4.4 ± 0.7	1304 ± 205	0.089 ± 0.008	0.253 ± 0.015	0.139 ± 0.006	0.117 ± 0.006	0.112 ± 0.039
GradNorm*	0.691 ± 0.013	7.8 ± 0.6	4.1 ± 0.4	1200 ± 122	0.092 ± 0.001	0.251 ± 0.012	0.138 ± 0.003	0.118 ± 0.007	0.093 ± 0.019
HOA-2	0.651 ± 0.029	63.0 ± 0.9	31.0 ± 0.5	9125 ± 140	0.091 ± 0.011	0.245 ± 0.002	0.135 ± 0.000	0.107 ± 0.003	0.074 ± 0.023
HOA-3	0.598 ± 0.029	63.0 ± 0.9	31.0 ± 0.5	9125 ± 140	0.083 ± 0.022	0.239 ± 0.007	0.127 ± 0.008	0.107 ± 0.003	0.043 ± 0.002
HOA-4	0.597 ± 0.015	63.0 ± 0.9	31.0 ± 0.5	9125 ± 140	0.094 ± 0.009	0.238 ± 0.002	0.115 ± 0.014	0.107 ± 0.003	0.043 ± 0.002
TAG-2	0.624 ± 0.015	17.4 ± 0.5	9.8 ± 0.3	2876 ± 88	0.083 ± 0.004	0.242 ± 0.005	0.134 ± 0.001	0.110 ± 0.007	0.055 ± 0.006
TAG-3	0.613 ± 0.032	20.5 ± 0.7	11.3 ± 0.2	3313 ± 56	0.094 ± 0.005	0.233 ± 0.002	0.122 ± 0.013	0.110 ± 0.008	0.055 ± 0.008
TAG-4	0.603 ± 0.027	25.2 ± 0.8	13.7 ± 0.3	4016 ± 80	0.083 ± 0.005	0.233 ± 0.002	0.122 ± 0.013	0.110 ± 0.008	0.055 ± 0.008
MAS-2	0.578 ± 0.015	8.8 ± 0.5	4.9 ± 0.3	1431 ± 94	0.069 ± 0.006	0.231 ± 0.006	0.124 ± 0.002	0.102 ± 0.003	0.052 ± 0.003
MAS-3	0.555 ± 0.015	9.7 ± 0.5	5.4 ± 0.3	1589 ± 94	0.072 ± 0.006	0.222 ± 0.006	0.124 ± 0.002	0.095 ± 0.003	0.042 ± 0.003
MAS-4	0.548 ± 0.001	12.9 ± 0.6	6.7 ± 0.3	1969 ± 75	0.070 ± 0.002	0.230 ± 0.008	0.111 ± 0.000	0.095 ± 0.007	0.042 ± 0.001

* All-in-one methods

Table 3: Comparison of test loss, training time, energy consumption, and carbon footprint on task set `sdnkt`.

Method	Total Loss	Time (h)	Energy (kWh)	CO2eq (g)	e	r	c	k	t
One-by-one	1.055 ± 0.034	23.0 ± 3.7	11.1 ± 2.2	3277 ± 660	0.148 ± 0.000	0.371 ± 0.029	0.386 ± 0.006	0.107 ± 0.003	0.043 ± 0.002
FedAvg*	1.130 ± 0.022	13.6 ± 0.8	5.0 ± 0.3	1478 ± 84	0.146 ± 0.001	0.379 ± 0.019	0.393 ± 0.002	0.110 ± 0.003	0.079 ± 0.013
FedProx*	1.101 ± 0.014	10.2 ± 0.3	6.2 ± 0.2	1818 ± 61	0.146 ± 0.001	0.369 ± 0.008	0.393 ± 0.001	0.113 ± 0.004	0.081 ± 0.012
GradNorm*	1.154 ± 0.055	10.4 ± 0.6	5.0 ± 0.2	1462 ± 70	0.147 ± 0.002	0.381 ± 0.015	0.394 ± 0.001	0.149 ± 0.062	0.082 ± 0.005
HOA-2	1.082 ± 0.032	82.6 ± 0.5	38.3 ± 0.3	11265 ± 86	0.149 ± 0.003	0.365 ± 0.025	0.394 ± 0.002	0.109 ± 0.002	0.064 ± 0.022
HOA-3	1.062 ± 0.024	82.6 ± 1.1	38.3 ± 0.2	11265 ± 53	0.149 ± 0.001	0.365 ± 0.014	0.394 ± 0.001	0.109 ± 0.006	0.046 ± 0.007
HOA-4	1.053 ± 0.034	82.6 ± 0.5	38.3 ± 0.3	11265 ± 86	0.148 ± 0.002	0.369 ± 0.028	0.386 ± 0.006	0.105 ± 0.001	0.045 ± 0.003
TAG-2	1.095 ± 0.033	26.5 ± 2.0	14.0 ± 0.9	4119 ± 279	0.147 ± 0.002	0.379 ± 0.013	0.393 ± 0.000	0.108 ± 0.005	0.068 ± 0.015
TAG-3	1.091 ± 0.034	28.2 ± 1.2	14.4 ± 0.6	4242 ± 170	0.147 ± 0.002	0.388 ± 0.014	0.396 ± 0.002	0.109 ± 0.009	0.050 ± 0.011
TAG-4	1.087 ± 0.028	34.6 ± 1.1	17.4 ± 0.5	5114 ± 159	0.147 ± 0.002	0.384 ± 0.011	0.396 ± 0.002	0.109 ± 0.009	0.050 ± 0.011
MAS-2	1.039 ± 0.024	13.0 ± 1.1	6.7 ± 0.2	1957 ± 53	0.143 ± 0.001	0.343 ± 0.014	0.393 ± 0.001	0.104 ± 0.006	0.056 ± 0.007
MAS-3	1.015 ± 0.018	14.2 ± 0.4	7.2 ± 0.2	2108 ± 50	0.143 ± 0.000	0.336 ± 0.005	0.383 ± 0.001	0.102 ± 0.008	0.052 ± 0.009
MAS-4	1.002 ± 0.014	14.8 ± 0.2	7.6 ± 0.0	2229 ± 14	0.143 ± 0.000	0.336 ± 0.005	0.383 ± 0.001	0.094 ± 0.009	0.046 ± 0.004

*All-in-one methods

Table 4: Comparison of test loss, training time, energy consumption, and carbon footprint on task set `erckt`.

Method	Total Loss	Time (h)	Energy (kWh)	CO2eq (g)	s	d	n	k	t	e	r	c	a
One-by-one	1.46 ± 0.011	31.0 ± 0.8	11.9 ± 0.5	3512 ± 151	0.08 ± 0.009	0.24 ± 0.014	0.10 ± 0.001	0.10 ± 0.002	0.04 ± 0.003	0.15 ± 0.001	0.35 ± 0.011	0.38 ± 0.002	0.02 ± 0.000
FedAvg*	1.49 ± 0.025	12.2 ± 0.3	4.9 ± 0.2	1435 ± 60	0.09 ± 0.002	0.23 ± 0.009	0.13 ± 0.002	0.10 ± 0.002	0.07 ± 0.005	0.14 ± 0.001	0.33 ± 0.011	0.39 ± 0.001	0.02 ± 0.001
FedProx*	1.49 ± 0.010	15.2 ± 0.3	7.3 ± 0.3	2151 ± 99	0.08 ± 0.000	0.23 ± 0.005	0.12 ± 0.001	0.10 ± 0.001	0.07 ± 0.010	0.14 ± 0.000	0.33 ± 0.006	0.39 ± 0.000	0.02 ± 0.000
GradNorm*	1.50 ± 0.049	12.2 ± 2.0	5.3 ± 1.3	1561 ± 377	0.08 ± 0.004	0.24 ± 0.014	0.13 ± 0.003	0.10 ± 0.003	0.07 ± 0.011	0.14 ± 0.001	0.34 ± 0.018	0.39 ± 0.001	0.02 ± 0.001
TAG-2	1.49 ± 0.025	30.3 ± 0.4	14.7 ± 0.8	4317 ± 229	0.09 ± 0.002	0.23 ± 0.008	0.13 ± 0.002	0.10 ± 0.002	0.07 ± 0.005	0.14 ± 0.001	0.33 ± 0.011	0.39 ± 0.001	0.02 ± 0.001
TAG-3	1.44 ± 0.014	34.5 ± 3.1	16.5 ± 2.6	4854 ± 751	0.09 ± 0.006	0.23 ± 0.009	0.12 ± 0.001	0.10 ± 0.002	0.03 ± 0.004	0.14 ± 0.000	0.33 ± 0.009	0.39 ± 0.001	0.02 ± 0.000
TAG-4	1.44 ± 0.007	34.9 ± 2.7	15.8 ± 2.4	4639 ± 717	0.07 ± 0.003	0.24 ± 0.002	0.11 ± 0.001	0.10 ± 0.002	0.03 ± 0.004	0.14 ± 0.000	0.35 ± 0.003	0.39 ± 0.001	0.02 ± 0.000
MAS-2	1.45 ± 0.021	14.6 ± 0.5	6.0 ± 0.1	1947 ± 175	0.08 ± 0.003	0.22 ± 0.008	0.12 ± 0.001	0.10 ± 0.001	0.06 ± 0.004	0.14 ± 0.000	0.32 ± 0.011	0.39 ± 0.001	0.02 ± 0.001
MAS-3	1.39 ± 0.030	15.7 ± 0.6	6.6 ± 0.4	1955 ± 104	0.07 ± 0.005	0.22 ± 0.008	0.12 ± 0.002	0.08 ± 0.002	0.05 ± 0.003	0.14 ± 0.001	0.32 ± 0.011	0.38 ± 0.001	0.02 ± 0.000
MAS-4	1.40 ± 0.027	17.9 ± 0.5	7.5 ± 0.3	2201 ± 94	0.06 ± 0.004	0.22 ± 0.008	0.12 ± 0.003	0.08 ± 0.002	0.05 ± 0.001	0.14 ± 0.001	0.32 ± 0.011	0.39 ± 0.001	0.02 ± 0.001
MAS-5	1.40 ± 0.028	20.0 ± 0.7	8.3 ± 0.4	2439 ± 105	0.06 ± 0.004	0.22 ± 0.008	0.12 ± 0.003	0.08 ± 0.002	0.05 ± 0.000	0.14 ± 0.002	0.32 ± 0.011	0.39 ± 0.001	0.02 ± 0.001

*All-in-one methods

Table 5: Comparison of test loss, training time, energy consumption, and carbon footprint on `sdnkterca`.