We thank the reviewers for valuable feedbacks. In below response, the reference order is the same as submitted work.

Reviewer # 1:

1) The convergence theory should be w.r.t $\sum_i f_i(\theta_i)$ and/or $\sum_i f_i(w)$, why care about $\|w-\theta_i\|$? $\rightarrow$ pFedMe can be written compactly as $\min_w \min_{\theta_i} (1/N) \sum_{i=1}^N \{f_i(\theta_i) + (\lambda/2)\|\theta_i - w\|^2\}$, which shows that there is dependency between solution components $\{\theta_i\}_i$ and w. Thus, we cannot separately evaluate each objective term. Similar to, e.g., l_2-regularization where $f_i(w) = l_i(w) + (\lambda/2)\|w\|^2$ for some loss function $l_i(w)$ and the overfitting regularization term $\|w\|^2$ plays a critical role in convergence of f_i, we believe that it makes sense to consider the personalized regularization $\|\theta_i - w\|^2$ in `pFedMe`'s convergence analysis. If we want to move far away from w, setting λ small is one option.

2) To what extent of λ to leverage client aggregation? From Theorem 1, $\lambda = 0$ to minimize $F(w) - F^*$? $\rightarrow$ The trade-off between personalization and exploiting data aggregation by varying λ is skilfully phrased by the reviewer. Note that $\lambda \in (0, \infty)$ avoids extreme cases of $\lambda = 0$ (no FL) and $\lambda = \infty$ (no personalization). The reviewer is correct in that setting $\lambda = 0$ reduces to simple ERM on each local client, but it is not true that in order to minimize $F(w) - F^*$, we should just set $\lambda = 0$, because Theorem 1 is based on using the Moreau envelope F_i, which implicitly requires $\lambda > 0$ by its definition [39]. In other words, if $\lambda = 0$, there would be no local update in line 8, and hence no Theorem 1 (and 2). On the other hand, similar to the standard way to find a "sweet spot" for hyper-parameters in l_1- or l_2-regularization, we fine-tune λ based on each dataset for each task in our experiments.

3) Are personalized model and global model plottings w.r.t θ_i and w, respectively? $\rightarrow$ Yes.

4) Use personalized models for Per-FedAvg's performance? $\rightarrow$ We did use $\theta_i(w) = w - \alpha \nabla f_i(w)$ for Per-FedAvg.

5) How to compare convergence rates with prior methods when they have different objectives? $\rightarrow$ The standard convergence analysis is performed according to the **loss functions' properties such as strong/non-convexity and/or smoothness** [49], regardless of different learning taks/objectives. Therefore, the quadratic speedup of pFedMe is compared with other linear-speedup methods having the same strong convexity and smoothness properties [3, 6, 19].

Reviewer #2:

1) Discuss with decentralized frameworks of Vanhaesebrouck, Bellet, Tommasi (2017) $\rightarrow$ We will update it.

2) How to choose λ to ensure strong convexity of h, considering the assumption on f_i is in expectation? $\rightarrow$ Assumption 1 can apply a stronger L-smoothness assumption on $\tilde{f}_i(\theta_i; \xi_i), \forall \xi_i$, instead of f_i. In this case, we can choose arbitrary $\lambda > 0$ for strongly convex and $\lambda > L$ for nonconvex loss functions, respectively.

Reviewer #4:

1) β larger than 1 could yield some instabilities (e.g., Fig 5) and inducing one more costly hyperparameter to tune? $\rightarrow$ In Theorem 1, we need $\eta \leq \frac{\hat{\eta}_1}{\beta R}$, which means larger β requires **smaller** η for the stability. In practice, we can fix β to an arbitrary constant, say 3, and perform the simple fine-tuning task for η.

Algorithm	MNIST	Synthetic
FedAvg	94.11 ± 0.05	77.69 ± 0.2
Per-FedAvg	94.22 ± 0.02	79.79 ± 0.09
pFedMe-GM	94.18 ± 0.06	78.65 ± 0.25
pFedMe-PM	**95.62** ± 0.04	**83.20** ± 0.06

Figure 1: Accuracy comparison with fine-tuned hyperparameters. $R = 20$, $K = 5$ for `pFedMe`, $R = 100$ for others.

2) Run $K * R$ local steps for Per-FedAvg and FedAvg. $\rightarrow$ It would be unfair to compare `pFedMe` with these using $K * R$ local steps. This is because for every K steps in a local round r, `pFedMe` only uses a single mini-batch, and thus only $\mathbf{R}$ mini-batches during R local rounds. On the other hand, Per-FedAvg uses 2 different mini-batches for 1 local update, and thus after $K * R$ local rounds, it will perform $\mathbf{2 * K * R}$ mini-batch updates. Similarly, FedAvg uses one mini-batch for 1 local update, so in total it will have $\mathbf{K * R}$ mini-batch updates. However, we still report an additional comparison when FedAvg and Per-FedAvg were trained over $K * R$ local steps in Fig. 1 here (in a strongly convex setting), which shows the personalized model (PM) of `pFedMe` still outperforms others.

3) Use a more realistic FL dataset than MNIST (LEAF benchmark)? $\rightarrow$ For synthetic data, we used a similar method to LEAF to generate the data. For real data, in LEAF, FEMNIST is distributed to a large number of clients, each with a small local dataset (of size around 226), which is not suitable for pFedMe because it needs a larger local dataset on each client (due to fresh mini-batch sampling in every round in line 7). Thus we use MNIST with a smaller number of clients, so each client can have up to 3,834 data samples. We note that the way we generate non-i.i.d and heterogeneous data using MNIST is similar to that using FEMNIST in LEAF.

4) How do (local) test performances computed and distributed? Any weighting? $\rightarrow$ W.r.t test accuracy of all algorithms, we sum all correctly classified samples over clients and divide it by the total number of samples of all clients without weighting. Their histograms are quite similar and we reported their means and standard deviation in Table 1.

5) The hyper parameters were optimised on the local test sets or on local validation sets? $\rightarrow$ On local test set. We have additionally fine-tuned hyper-parameters on a validation set, and still obtain the same values as those on the test set.

6) Why is global model of pFedMe is lagging in Fig. 2? Due to varying number of clients or lower sampling rate? $\rightarrow$ The **non-fine-tuned hyper-parameter** is the main reason for the lagging performance of the global model (GM) in Fig. 2. In cases where hyperparameters are fine-tuned, the GM is consistent and performs well compared to FedAvg (see Table 1). We have run more experiments and observed that the GM performance is still consistent with varying numbers of clients and sampling rates. The result is not presented here due to the lack of space.

7) Finally, we thank the reviewer for correcting several typos.