Thank you to all the reviewers for the feedback! R2 and R5 are positive about the paper and gave very helpful suggestions. We believe that R4 may have a misunderstanding about the algorithm and analysis, and we have tried to clarify it below. We hope that R4 will reconsider and increase their score to recommend acceptance. Thanks again!

[R4] Does FedNova decrease the stepsize of each client and slow down the convergence speed? `FedNova` does not degrade the convergence speed of `FedAvg`, as we justify next. Firstly, when clients perform local updates, they use the same step size η as `FedAvg` and not a scaled one $\tau_{\text{eff}}\eta/\tau_i$. Secondly, the difference between `FedNova` and `FedAvg` is the aggregation weights w_i, which only control the direction, and not the magnitude of the accumulated global update. The magnitude is determined by τ_{eff}, which is the same in `FedAvg` and `FedNova`. For example, observe in Figure 1 that the solid green vector $(\boldsymbol{x}_{\text{FedNova}}^{(t+1,0)} - \boldsymbol{x}^{(t,0)})$ has roughly the same magnitude as the `FedAvg` update $(\boldsymbol{x}_{\text{FedAvg}}^{(t+1,0)} - \boldsymbol{x}^{(t,0)})$.

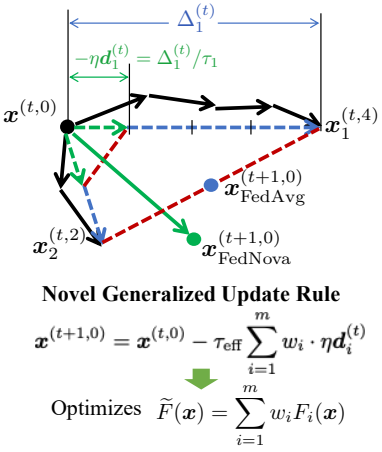

Figure 1: The difference between `FedAvg` and `FedNova` is the aggregation weights w_i, which only controls the direction of the solid green arrow.

[R4 & R2] What if we force all clients to run the same local steps (*e.g.*, the minimum of local steps across clients)? It is true that forcing all clients to perform $\tau = \min_i \tau_i$ local steps (let us call this algorithm `FedAvg-min`) can also ensure objective consistency. However, its convergence rate is *provably worse* than `FedNova`. This is because, in each round, `FedAvg-min` will go over less data samples than `FedNova` ($mb\tau_{\min}$ versus $b\sum_{i=1}^m \tau_i$ where b is the mini-batch size). Using theorem 2, one can show that the convergence rate of `FedAvg-min` is $1/\sqrt{mT\min_i \tau_i}$, which is slower than the rate of `FedNova` $1/\sqrt{T\sum_{i=1}^m \tau_i}$. Empirically, we evaluate the performance of `FedAvg-min` and `FedNova` on the synthetic dataset in Figure 2. Observe that `FedNova` achieves lower loss value than `FedAvg-min` at any round. Another drawback of a fixed τ algorithm like `FedAvg-min` is that faster nodes would remain idle in each round while waiting for slower nodes. `FedNova` avoids such straggling delays by allowing nodes to make different numbers of local updates.

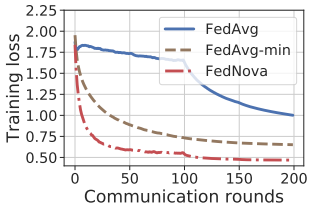

Figure 2: `FedAvg-min`.

[R4] Do we need to know $\tilde{\tau}$ in order to choose a suitable η in Theorem 3? We do not need to know $\tilde{\tau}$ or τ_i beforehand. It is worth noting that $m\tilde{\tau}T = \sum_{i=1}^m \sum_{t=0}^{T-1} \tau_i^{(t)}$ is actually the total number of processed mini-batches across all clients after T rounds. Once we have a budget on the total mini-matches $K = m\tilde{\tau}T$ to be processed, we can set the learning rate as $\eta = \sqrt{m/\tilde{\tau}T} = \sqrt{m^2/K}$, then the optimization error is guaranteed to be bounded by $\mathcal{O}(1/\sqrt{K})$. As for the upper bound on learning rate, it is only used for theoretical analysis. In practice, one always needs to tune the learning rate.

[R4] Is the general analysis framework a marginal contribution? We believe that the analytical framework proposed in Section 4 is an important and impactful contribution, perhaps even more critical than Section 5. This is because: 1) we identify the objective inconsistency problem in `FedAvg` by showing that performing the same number of local epochs at clients with heterogeneous sizes datasets optimizes a mismatched objective and 2) we provide the first (to the best of our knowledge) rigorous understanding of the objective inconsistency problem in federated learning by quantifying the non-vanishing gap caused by incorrect weighted aggregation of heterogeneously updated models.

[R4] Extending our theorems to strongly convex case. We focus on the non-convex case since it is the most practical and challenging setting. It is straightforward to extend our analysis to convex or strongly convex cases. For instance, one can directly apply Polyak-Łojasiewicz condition to Eqn. (89) in the appendix and obtain an improved rate.

[R2] Is the bias-correction necessary when using local momentum? Yes, it is necessary because without bias-correction, the algorithm will converge to a stationary point of a mismatched objective, the analytical form of which can be derived using our framework. We will add some experiments in appendix to further validate this.

[R5] Clarifications on `FedProx`: hyper-parameters and differences to `FedNova`. On the synthetic dataset, we use the same model and hyper-parameters as the `FedProx` paper. We set $\mu = 1$ because this is the best value reported in that paper. On the CIFAR-10 dataset, we tuned the value of μ from $\{0.0005, 0.001, 0.005, 0.01\}$ as stated in the Appendix. `FedNova` with proximal updates is same as `FedProx` in terms of the local updates, but the aggregation weights w_i and effective steps τ_{eff} are set differently. In our framework (4), the weights and τ_{eff} used in `FedProx` are given by Eqn. (6) while `FedNova` uses $w_i = p_i$ and $\tau_{\text{eff}} = \sum_{i=1}^m p_i\tau_i$.

[R5] Avoiding confusions on the algorithm name. Thanks for the suggestion! We will avoid using the term 'normalized gradient' and clearly state the meaning of normalization in our paper, or even use another term.