Data-Free Contribution Estimation in Federated Learning using Gradient von Neumann Entropy

Supplementary Material

A. Rank Adaptive Kalman Filter Algorithm

Algorithm 3 `Rank Adaptive Kalman Filter`

Initialize: x_i^0, P^0, Q, ϵ
Input: $\hat{x}_i^{(t-1)}, s_i^{(t)}, \gamma_i^{(t)}$
1: **Predict:** $\hat{x}_{i|t-1}^{(t)}$ using Equation 5
2: **Update**:
3: $\rho_s^{(t)} = \texttt{Spearman}\,(\hat{\mathbf{x}}_{|t-1}^{(t)}, \mathbf{s}^{(t)})$
4: $\rho_\gamma^{(t)} = \texttt{Spearman}\,(\hat{\mathbf{x}}_{|t-1}^{(t)}, \boldsymbol{\gamma}^{(t)})$
5: $y_i^{(t)} = (s_i^{(t)}, \gamma_i^{(t)})^\top$
6: Update $R_i^{(t)}$ from Equation 17
7: Compute: $\hat{x}_i^{(t)}$ using Equation 7
8: **return** $\hat{x}_i^{(t)}$

We provide an extended analysis of the RAKF proposed in section 4.3 here. The filter fuses two complementary, data-free signals :(i) the spectral (von Neumann) entropy of the final-layer update and (ii) the class-specific Shapley alignment (CSSV), into a single per-client contribution estimate that is stable over rounds yet responsive to persistent changes. Algorithm 3 complements Figure 3 to illustrate the filter methodology.

We demonstrate the filter operation process in Figure 5 by analyzing the internal values of the filter. In Figure 5 (a), the filter initially tracks the correlation trajectory of the entropy and later transitions to CSSV. Figures 5 (b-c) illustrate this: we observe that while the entropy-based rankings are initially consistent, they exhibit greater variation between rounds 50-150. This, in turn, reduces its variance (Figure 5 (d)), and therefore the fused weights are more dependent on the CSSV signal. An key caveat is that the filter does not have access to ground-truth correlations, as the client utility (standalone accuracy) is unknown to the server. Therefore, it may trust a signal as long as it is consistent, even if it may have a lower actual correlation. This is evident in Figure 5 (a), where the signal with lower correlation (CSSV) is more trusted due to its consistency despite having a lower true correlation score. Together, these plots illustrate the intended behavior of the filter, which initially reduces rank volatility but adapts to the signals when a sustained regime change occurs.

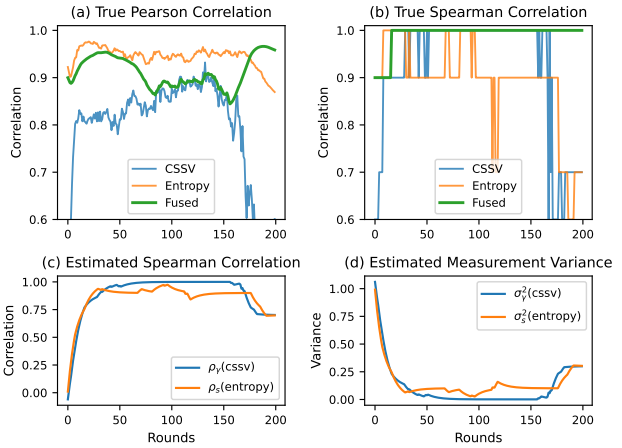


Figure 5. The RAKF's working principle demonstrated for a run on the Dirichlet ($\alpha = 0.1$) split of CIFAR-10. (a) The RAKF fused weights smoothen and track the true Pearson correlation trend of the perceived reliable signal. (see 'Fused' vs. 'CSSV') (b) Fused weights resist rank changes and therefore maintain a stable Spearman Rank correlation profile over rounds. (c) The filter estimates the rank correlation between its previous state and each incoming signal. (d) The signal that has a higher estimated rank correlation has lower variance and is considered more trustworthy.

B. Filter hyperparameter sensitivity

To assess the impact of filter hyperparameters Q and ϵ, on weight correlations with standalone accuracies, we provide a brief analysis of their sensitivity.

Process noise covariance Q. In Figure 6, we tested the impact of varying the process noise covariance Q on correlation performance over rounds. In order to interpret the results, we revisit the definition of Q given in Equation 14. Q is the estimate of the covariance of the normally distributed process noise in the state model. In a sense, it is a prior estimate of the fidelity of the process model that the filter is operating under. The better a process is modeled, the lower the value of Q should be. In our case, the process model is given as $x_i^{(t)} = x_i^{(t-1)} + w_i^{(t)}$, where $x_i^{(t)}$ is the estimate of the weights of the client i in round t. This implies that the client weights should remain static. In principle, this is a sound model as the clients' utility is expected to remain constant under the assumption that no client adds any additional data during the training process. However, given that the filter initializes under a uniform weight for each client, the client state should be allowed to change based on external heuristics. The choice of process noise covariance Q

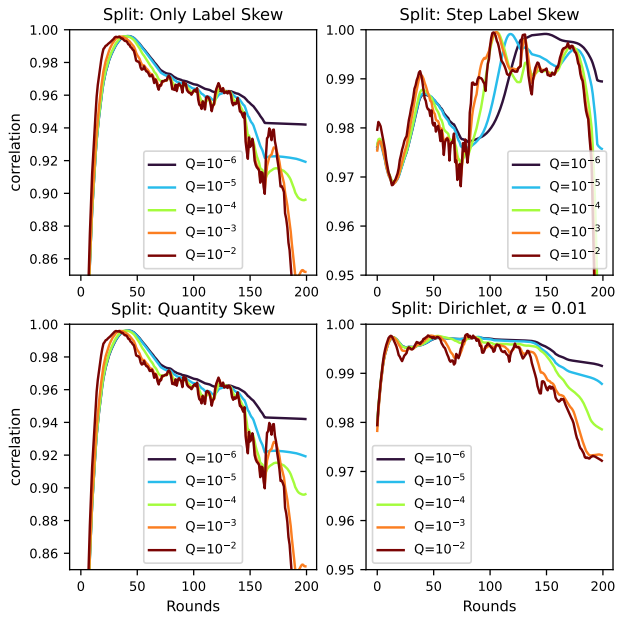


Figure 6. Effect of choice of filter process noise covariance Q on the per-round Pearson correlation with standalone accuracies for different splits of the CIFAR-10 dataset. Lower values of Q (see $Q = 10^{-6}$) yield smoother but stagnant curves as the filter 'trusts' its internal process model of persistent client ranks more.

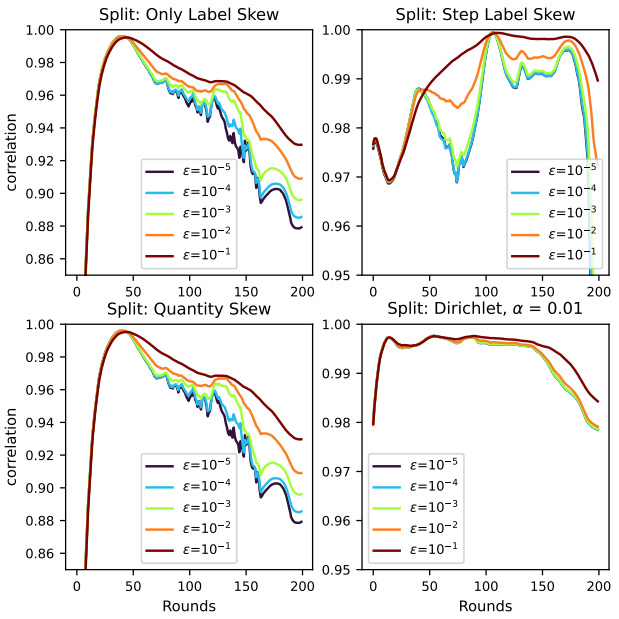


Figure 7. Effect of choice of measurement variance floor ϵ on the per-round Pearson correlation with standalone accuracies for different splits of the CIFAR-10 dataset. Higher values of ϵ raise the variance floor for the measurements (entropy and CSSV) and cap the filter's confidence in the measurement signals. The fused weights do not respond to rapid changes in the signals, resulting in a smoother correlation curve (see $\epsilon = 10^{-2}$)

governs how fast the state should respond to external signals when they differ from the internal state. We observe this phenomenon in practice in Figure 6. We observe that lower values of Q smooth the correlation curves and avoid the noise in the external signals (entropy and CSSV). Although this is a desirable behavior if the initial correlation is high (as in the Only Label Skew and Quantity Skew cases), it can be problematic when early correlations are poor and improve in later rounds (as in the Step Label Skew case). An intermediate value of Q provides a good balance between these behaviors. Across most splits, a broad middle band of values yields a similar performance, indicating low sensitivity in practice.

Measurement variance floor ϵ. Similarly to the study of the sensitivity of the process noise covariance, we study the impact of the choice of ϵ. To recap, the measurement variance floor provides a lower bound to the measurement noise covariance matrix R as described in Equation 17. The measurement noise covariance acts as a counterweight to the process noise covariance Q. In a typical Kalman Filter design, it provides a fixed prior on the covariance of external measurements. In the case of RAKF, $R^{(t)}$ is time-dependent and varies inversely with the rank correlation of the measurement signals (Equation 17). Intuitively, a higher value of R implies a relatively lower confidence in the external measurements as compared to the process model. Higher ϵ

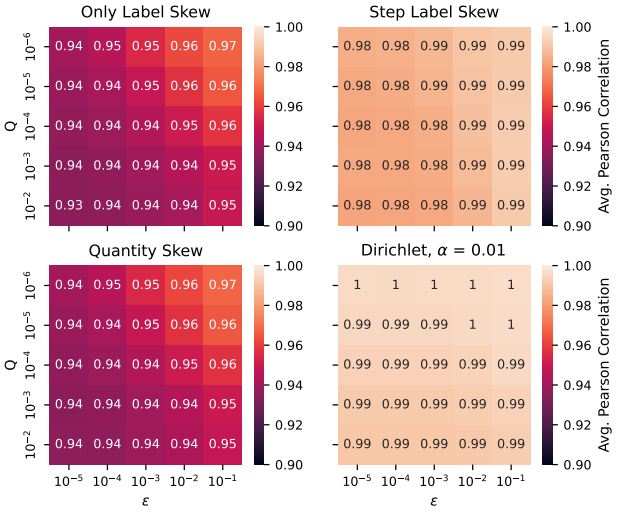


Figure 8. Sensitivity of choice of Q and ϵ on the average Pearson correlation of the fused weights and standalone accuracies for different splits on the CIFAR-10 dataset. Average values of correlation are robust with respect to changes in filter hyperparameters.

results in a generally higher value of R. Figure 7 captures the effect of changing ϵ on the true Pearson correlation over rounds for different splits of the CIFAR-10 dataset. The em-

pirical results validate this hypothesis, as higher values of ϵ yield a delayed, more sluggish response to entropy/CSSV.

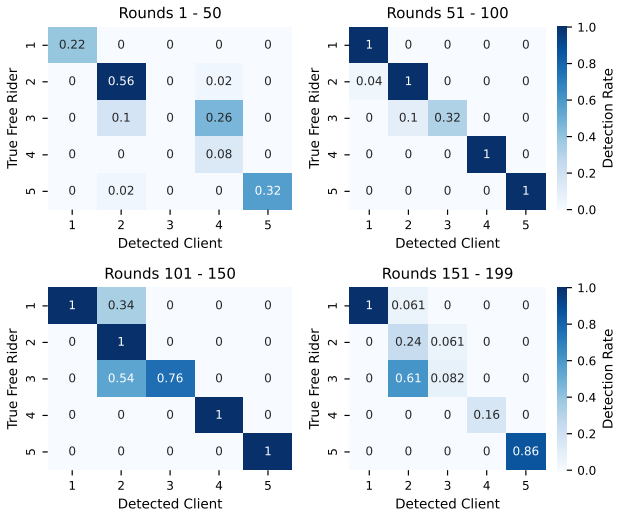
Although the choice of Q and ϵ affects the behavior and performance of the RAKF, the variation observed in the correlation values when aggregated over rounds remains negligible. We capture this in Figure 8, which shows the variation in the average correlation values as a function of (Q, ϵ). Correlations remain high over wide regions of the grid for all splits, with very small variations for all splits. Practically, one can pick Q to target desired smoothness and set a modest ϵ to avoid over-trusting rapidly fluctuating signals. We therefore show that the reported results are reasonably robust to the choice of filter hyperparameters.

C. Free-rider detection

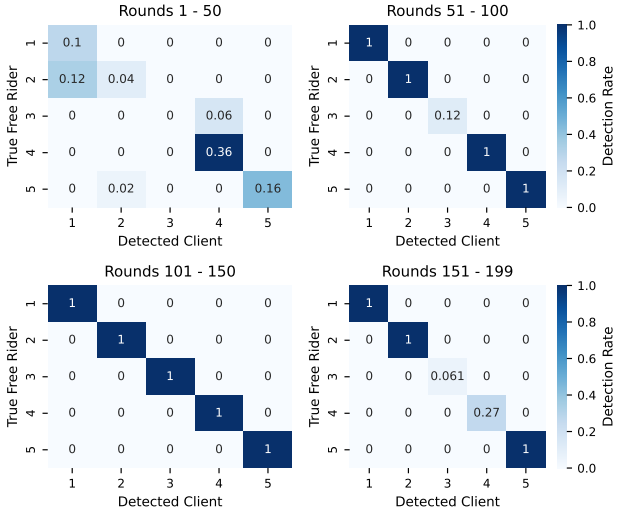
Contribution estimation schemes should not only be effective in identifying client utility under data heterogeneity but also should be able to detect scenarios where there are non-contributing clients or free-riders. A free-rider could be modeled in several ways. In the trivial scenario, it could be a client that does not perform training but participates only to obtain the global model or participation rewards. At the other extreme, a free rider could be malicious and attempt to actively derail the FL process. The first scenario can be easily identified by a simple client gradient norm check. The latter model requires further assumptions about a white-box, grey-box, or black-box knowledge setting and is better suited for treatment under the robust FL paradigm. We simulate a non-adversarial free-rider that participates by duplicating a very small amount of data to match the average cohort data size. Any method that correlates with the amount of data only would not be able to distinguish this free-rider.

In order to detect the free-rider from the contribution estimates, we first transform the estimates to the real space (contribution estimates are compositional as they have a fixed sum, one, and thus lie on a simplex). We then compute the robust Z-score $z_i = (x_i - \text{median}(x_i)/\text{MAD})$ where MAD is the Median Absolute Deviation. Free-riders are clients that are flagged as outliers in the Z-score. We run 5 experiments on the Dirichlet-partitioned (α = 1.0) CIFAR-10 dataset for both SpectralFed and SpectralFuse. One out of the 5 clients is the free rider in each experiment. We show the fraction of detections for each client in four 50-round intervals in Figure 9.

For both SpectralFed (Fig. 9a) and SpectralFuse (Fig. 9b), free-riders are detected with high accuracy towards the middle and later phases of training. We observe lower early-phase detection with occasional off-diagonal false positives, likely because the weights are not yet well-separated. These results indicate that simple thresholding of our data-free



(a) SpectralFed



(b) SpectralFuse

Figure 9. Free-rider detection accuracy of the proposed methods at four different intervals of training. Each row denotes an independent federated training procedure, with the y-axis indicating which client is the free rider. The values in the grid represent the rate of detecting the client, denoted on the x-axis as the free rider. High values on the diagonal indicate correct detections, while off-diagonal values indicate false positives.

scores is effective for free-rider screening, without the need for auxiliary validation data or self-reported metadata.

D. Choice of final layer for entropy evaluation

In this section, we motivate the choice of using the final layer of the gradients for the entropy computation with empirical evidence. Figure 10 shows the average Pearson correlation between entropy and standalone accuracy as a function of layer depth for a 5-layer CNN on CIFAR-10 and

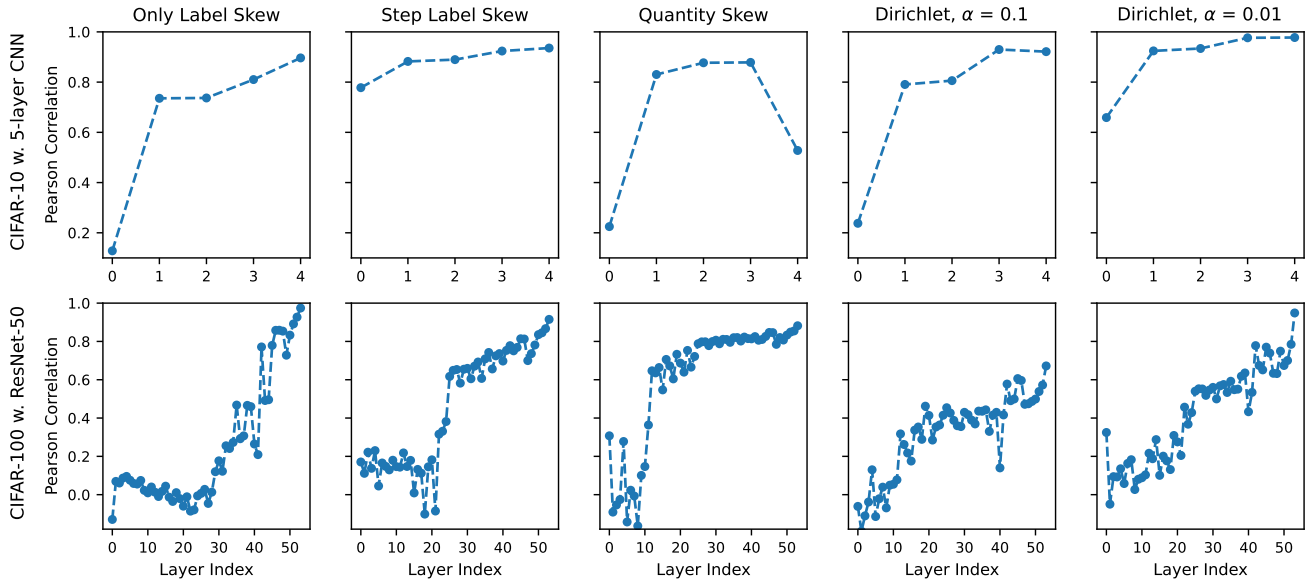


Figure 10. Average Pearson correlation of the entropy of the gradient for each layer with the standalone accuracy. The top row displays the average correlation for the 5-layer CNN used on the CIFAR-10 dataset, with each entry corresponding to a specific split. The x-axis denotes the layer index. The bottom row shows a similar analysis for the ResNet-50 model used on the CIFAR-100 dataset. We observe a general trend of increasing correlation as we move deeper in the network, with the final layer usually showing the highest correlation.

ResNet-50 on CIFAR-100. We observe a generally increasing correlation as we move deeper in the model with the highest correlations observed in the final layer. This provides strong empirical grounds for choosing the final layer only, rather than, e.g., averaging across layers Furthermore, these results align with the theoretical basis for the role of the classifier head (final layer) in encoding the class geometry, as discussed in subsection 4.1. When a client's data is uniform and diverse in labels, the final layer update exhibits a richer and more balanced spectrum (higher entropy), which better tracks client utility.

The choice of using the last layer for entropy computation motivates the use of Class-wise Shapley Values (CSSV) as an auxiliary signal for fusion. As discussed in subsection 4.2, CSSV is also computed on the last layer. Therefore, both the metrics used as inputs for fusion are derived from the same sub-part of the model, albeit with different approaches. Additionally, computing entropy on the last layer is the most efficient, as the Gram matrix $A \in \mathbb{R}^{C\times C}$ (see subsection 4.1) and scales only with the number of classes.

E. Compute Complexity

For n participating clients, let the final-layer update of each client be $M_i \in \mathbb{R}^{d\times C}$, where d is the feature dimension and C is the number of classes, with typically $d > C$. For each client, we first form the class-space Gram matrix, $A_i = M_i^\top M_i \in \mathbb{R}^{C\times C}$, which costs $O(d\,C^2)$. We then compute the eigenspectrum of A_i using SVD, which costs $O(C^3)$, and evaluate the entropy from the normalized eigenvalues in $O(C)$ time. Therefore, the total time complexity of entropy computation over all participating clients is

$$O\big(n\,d\,C^2 + n\,C^3\big)\,. \tag{18}$$

Because these computations are performed independently per client, they are embarrassingly parallel across clients. The additional cost of SPECTRALFUSE relative to SPECTRALFED comes from computing CSSV and applying the Kalman filter.

We report the per-round *server-side* wall-clock time for the baselines and our methods in Table 3. The reported values are mean ± standard deviation in milliseconds over 10 rounds with all methods evaluated under identical CPU/GPU settings.

Table 3. Per-round server-side wall-clock time (ms) for running contribution estimation and aggregation.

Dataset	CGSV	ShapFed	SpectralFed	SpectralFuse
CIFAR-100	50 ± 2	467 ± 31	207 ± 2	378 ± 12
FedISIC	172 ± 6	398 ± 49	53 ± 10	102 ± 5

Table 3 shows that SPECTRALFED is substantially cheaper than ShapFed in both settings. The entropy-based method are the fastest on FedISIC as they scale only with the number of classes. In contrast, the time complexity of CGSV scales with the full model size resulting in slower runtimes for larger models.