Fast-FedUL: A Training-Free Federated Unlearning with Provable Skew Resilience

Thanh Trung Huynh[‡1], Trong Bang Nguyen[‡2], Phi Le Nguyen[2], Thanh Tam Nguyen[3], Matthias Weidlich[4], Quoc Viet Hung Nguyen[3], and Karl Aberer[1]

[1] Ecole Polytechnique Federale de Lausanne, Switzerland
[2] Hanoi University of Science and Technology, Vietnam
[3] Griffith University, Australia
[4] Humboldt-Universität zu Berlin, Germany

A Appendix

A.1 Notation Summary

Symbols	Definition
S	central server S
$\{e_1, e_2, ..., e_N\}$	set of all clients
$\{D_1, D_2, ..., D_N\}$	set of all datasets of clients
$\mathcal{M}_t$	global model at training round t
$\Delta\mathcal{M}_t^i$	update of client e_i at round t
α	Lipschitz coefficient
ϵ_t^i	local skew for client e_i at training round t
$\Delta_t = \mathcal{M}_t^* - \mathcal{M}_t$	Difference between re-trained global model and original global model
$tr(X)$	trace of matrix X

A.2 End-to-end Federated Learning and Unlearning Process

Algorithm 1 outlines our federated unlearning process integrated into an ongoing continuous federated learning (FL) pipeline. The FL process initiates by initializing the global model (Line 1) and then proceeds to iteratively distribute its training across client devices (Line 2-10). Within each training round, a subset of clients is efficiently selected using our sampling methods (Line 3). Subsequently, the chosen clients retrieve the global model (Line 5), locally train it with their respective data for a set number of iterations (Line 6-7), and transmit their local model gradients to the server (Line 8) for aggregation (Line 9). The updates from the sampled clients in each round are aggregated and stored at the server (Line 10), facilitating the federated unlearning process. Upon receiving an unlearning request from a user client e_k (Line 11), our proposed efficient unlearning technique (as detailed in Alg. 1) removes their contributions up to the current round from the global model; and the client is excluded from subsequent training steps.

[‡] Both authors contributed equally to this research.

Algorithm 1: FL with Unlearning

input : Clients $E = \{e_1, ..., e_N\}$ with local data $\{D_1, ..., D_N\}$;
number of training iterations T_t;
number of sampled clients m
output: Global model $\mathcal{M}$

Initialize the global model $\mathcal{M}_0$
for $t \in \{1, \ldots, T\}$ **do** // For each iteration
  Sample C^t from E;
  for $e_i \in C^t$ **do** // For each sampled client
    Download $\mathcal{M}_{t-1}$ from server to $\mathcal{M}^i_{t-1}$
    for $k \in \{0, \ldots, R-1\}$ **do** // For each epoch
      $\Delta\mathcal{M}^i_t = local\_train(\mathcal{M}^i_{t-1}, \mathcal{D}_i)$;
    Send $\Delta\mathcal{M}^i_t$ to server;
  $\mathcal{M}_t = \mathcal{M}_{t-1} + Agg(\{\Delta\mathcal{M}^i_{t-1} \mid e_i \in C^t\})$
  Store the updates from sampled clients C^t
  if *unlearning request of user* e_k **then**
    $\mathcal{M}_t = unlearning(\mathcal{M}_t, e_k, \alpha)$ // Unlearn the target client from the model if requested
return $\mathcal{M}_T$;

Algorithm 2: Unlearning Process.

input : Central server S; original global model $\mathcal{M}_T$; clients $E = \{e_1, ..., e_N\}$ with data $\{D_1, ..., D_N\}$; target client e_u; number of iterations T; number of sampled clients N, Lipschitz coefficient α.
output: Global model after unlearning $\mathcal{M}^*_T$.

$\Delta'_0 = 0$; // Initialize the model difference
for $t \in \{1, \ldots T\}$ **do** // Compute the model difference iteratively
  $\Delta'_t = (1+\alpha)\Delta'_{t-1} + \frac{1}{N(N-1)} \sum_{e_i \in C^{t-1}_F} \Delta\mathcal{M}^i_{t-1} - \frac{1}{N}\Delta\mathcal{M}^u_{t-1}$;
$\mathcal{M}'_T = \mathcal{M}_T + \Delta'_T$;
return $\mathcal{M}'_T$;

A.3 Extended Experiments

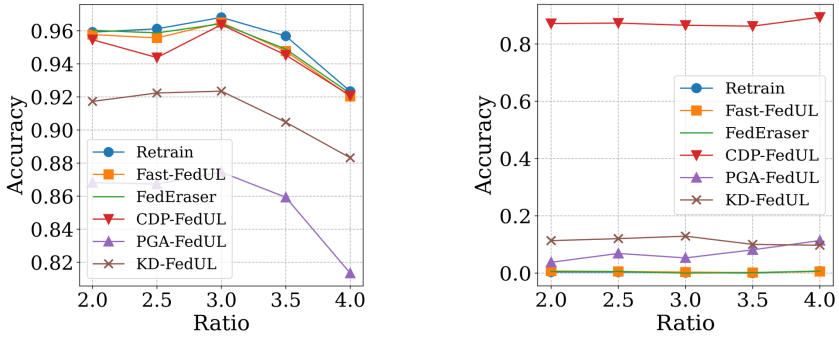
End-to-end Comparison. Tab. 1 reports the accuracy of the global on the *main task* and the *backdoor task* using the unlearning techniques against *pixel backdoor attack*. Similar to the *edge-case backdoor* scenario, our approach restores the model's efficacy in the main task and entirely neutralizes the attack's impact. Following Fast-FedUL's unlearning process, the global model maintains high accuracy levels of 98.47%, 94.13%, and 87.52% across the three datasets for the main task, which are nearly identical to the retraining model's performance. Furthermore, our technique efficiently mitigates the attack's threat, as evidenced by a success attack rate of less than 0.1% across all three datasets.

Robustness to Data Distribution. Fig. 1 presents the resilience of the methods when confronted with non-IID data in the context of the *pixel backdoor*

Table 1: End-to-end comparison with **pixel backdoor**.

Dataset	Pre-unlearned		Retrain		Fast-FedUL		FedEraser		CDP-FedUL		PGA-FedUL		KD-FedUL	
	main	backdoor	main	backdoor	main	backdoor	main	backdoor	main	backdoor	main	backdoor	main	backdoor
MNIST	0.9878	0.9220	0.9875	0.0036	0.9847	0.0018	0.9878	0.0036	0.9878	0.9056	0.8092	0.0346	0.9876	0.0780
CIFAR10	0.9657	0.8913	0.9520	0.0092	0.9413	0.0132	0.9474	0.0096	0.9463	0.7952	0.7683	0.0273	0.9467	0.0529
OCTMNIST	0.8837	0.8869	0.8847	0.0083	0.8752	0.0073	0.8551	0.0240	0.8810	0.8729	0.7844	0.0137	0.8550	0.0475

attack. Similar to the situation in the *edge-case attack* scenario, all approaches experience a decline in model effectiveness on the main task as the non-IID ratio increases. In this context, our Fast-FedUL and FedEraser techniques remain standout performers, demonstrating comparable quality to that of the retraining model. Conversely, PGA-FedUL exhibits significant vulnerability in non-IID scenarios, whereas CDP-FedUL consistently performs inadequately in mitigating backdoor attacks, as observed across various scenarios.



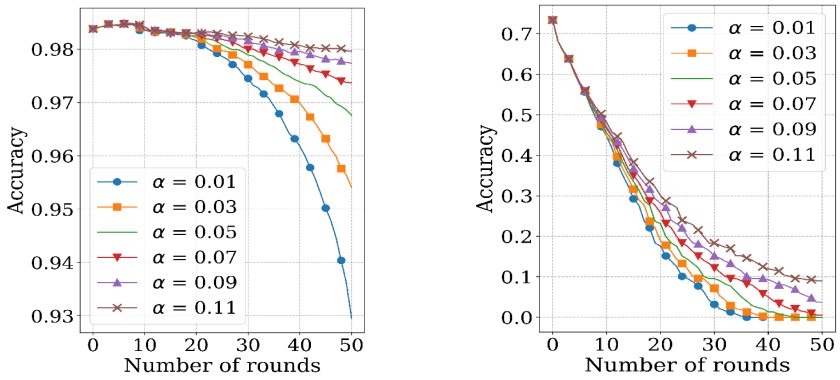
(a) Main task $\mathcal{T}_m$ (b) Backdoor task $\mathcal{T}_b$

Fig. 1: Robustness against non-IID data (*pixel backdoor*).

Hyper-parameter Sensitivity. We explore the effect of the Lipschitz coefficient α on the performance of the model, using the MNIST dataset with α ranging from 0.01 to 0.11. For each α, we report the accuracy of the main task and backdoor task of the model after each unlearning iteration.

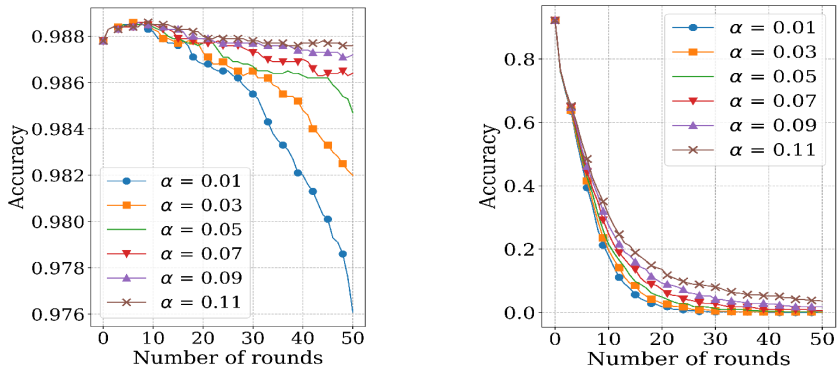
The results in Fig. 2 indicate that a small change of α can lead to a considerable change in the final model's accuracy on both tasks. This is due to the skew in the model accumulating over the iterations of the unlearning process. Based thereon, we recommend the coefficient to be set in the range of 0.05 to 0.1.

Fig. 3 depicts the sensitivity of our technique to the Lipschitz coefficient α under the *pixel backdoor attack* scenario. This outcome aligns with observations from the *edge-case backdoor attack*, indicating that the coefficient should ideally fall within the range of 0.05 to 0.1.



(a) Main task $\mathcal{T}_m$ (b) Backdoor task $\mathcal{T}_b$

Fig. 2: Effects of Lipschitz coef. α (**edge-case backdoor**).



(a) Main task $\mathcal{T}_m$ (b) Backdoor task $\mathcal{T}_b$

Fig. 3: Effects of Lipschitz coeff. α (**pixel backdoor**).

A.4 Theoretical Remarks

Lemma 1. *Let $\zeta_1, \zeta_2, \ldots, \zeta_N$ be vectors in $\mathbb{R}^d$ and $w_1, w_2, \ldots, w_N$ be non-negative numbers and $\sum_{i=1}^N w_i = 1$, C be a proper sampling. If $v \in \mathbb{R}^N$ is such that*

$$\mathbf{P} - pp^\top \preceq \mathrm{Diag}\left(p_1 v_1, p_2 v_2, \ldots, p_N v_N\right) \tag{1}$$

then

$$\mathbb{E}\left[\left\|\sum_{e_i \in C} \frac{w_i \zeta_i}{p_i} - \sum_{i=1}^N w_i \zeta_i\right\|^2\right] \leq \sum_{i=1}^N w_i^2 \frac{v_i}{p_i} \left\|\zeta_i\right\|^2,$$

where $\mathbb{E}$ is the expectation taken over C.

Applying the lemma to Eq. 4 in main text, with $w_i = \frac{1}{N}$ and $\zeta_i = \Delta \mathcal{M}_t^i$, we have Eq. 5 in main text.

Proof 1 - Optimal choice for v_i

Proof. From condition (1), we have:

$$D = \text{Diag}\left(p_1 v_1, p_2 v_2, \ldots, p_N v_N\right) - \left(\mathbf{P} - pp^\top\right) \succeq 0$$

It is equivalent to $\forall z \in \mathbb{R}^N$:

$$z^\top D z \geq 0$$

Consider $e_i = [0, 0, ..., 1, 0, .., 0] \in \mathbb{R}^N$, where only i-th element of e_i equals to 1. Then we have:

$$p_i(v_i - 1 + p_i) = e_i^\top D e_i \geq 0$$

It implies that $v_i \geq 1 - p_i$

Proof 2 - Optimal bound for objective function

Proof. Our proof technique can be seen as an extended version of that in [1] (Horváth & Richtárik, 2019). Let $1_{i\in C} = 1$ if $i \in C$ and $1_{i\in C} = 0$ otherwise. Likewise, let $1_{i,j\in C} = 1$ if $i, j \in C$ and $1_{i,j\in C} = 0$ otherwise. Note that $\mathrm{E}\left[1_{i\in C}\right] = p_i$ and $\mathrm{E}\left[1_{i,j\in C}\right] = p_{ij}$. Next, let us compute the mean of $X := \sum_{i\in C} \frac{\Delta\mathcal{M}_t^i}{p_i}$:

$$\mathrm{E}[X] = \mathrm{E}\left[\sum_{i\in C} \frac{\Delta\mathcal{M}_t^i}{p_i}\right] = \sum_{i=1}^N \frac{\Delta\mathcal{M}_t^i}{p_i} \mathrm{E}\left[1_{i\in C}\right] = \sum_{i=1}^N \Delta\mathcal{M}_t^i$$

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a matrix where $A_{ij} = tr\left(\frac{\Delta\mathcal{M}_t^{i\top}}{p_i} \frac{\Delta\mathcal{M}_t^j}{p_j}\right)$, and let e be the vector of all ones in $\mathbb{R}^N$. We now write the variance of X in a form which will be convenient to establish a bound:

$$\begin{aligned}
\mathrm{E}\left[\|X - \mathrm{E}[X]\|^2\right] &= \mathrm{E}\left[\|X\|^2\right] - \|\mathrm{E}[X]\|^2 \\
&= \mathrm{E}\left[\left\|\sum_{i\in C} \frac{\Delta\mathcal{M}_t^i}{p_i}\right\|^2\right] - \|\sum_{i=1}^N \Delta\mathcal{M}_t^i\|^2 \\
&= \mathrm{E}\left[\sum_{i,j} \boldsymbol{A}_{ij} 1_{i,j\in C}\right] - \|\sum_{i=1}^N \Delta\mathcal{M}_t^i\|^2 \\
&= \sum_{i,j} p_{ij} \boldsymbol{A}_{ij} - \sum_{i,j} tr\left(\Delta\mathcal{M}_t^{i\top} \Delta\mathcal{M}_t^j\right) \\
&= \sum_{i,j} \left(p_{ij} - p_i p_j\right) \boldsymbol{A}_{ij} \\
&= e^\top \left(\left(\boldsymbol{P} - pp^\top\right) \circ \boldsymbol{A}\right) e.
\end{aligned}$$

Since, by (1), we can further bound

$$e^\top \left(\left(\boldsymbol{P} - pp^\top\right) \circ \boldsymbol{A}\right) e \leq e^\top \left(\boldsymbol{Diag}(p \circ v) \circ \boldsymbol{A}\right) e = \sum_{i=1}^N p_i v_i \boldsymbol{A}_{ii}$$

From those, we have:

$$\mathrm{E}\left[\|X - \mathrm{E}[X]\|^2\right] \leq \sum_{i=1}^N p_i v_i \boldsymbol{A}_{ii} = \sum_{i=1}^N \frac{v_i}{p_i} \left\|\Delta\mathcal{M}_t^i\right\|^2 \tag{2}$$

Consider the case of independent sampling, then $\forall i \neq j : p_{ij} = p_i p_j$. It is equivalent to:

$$\boldsymbol{P} - pp^\top = \boldsymbol{Diag}(p \circ (1 - p))$$

For the optimal choice of v_i, the equality in (2) holds for independent sampling.

Proof 3 - Solution for optimal sampling By Lemma 1, the independent sampling is optimal. In addition, for independent sampling, (2) holds as equality. We have:

$$\alpha_C := \mathrm{E}\left[\sum_{i=1}^{N} \frac{1-p_i}{p_i} \left\|\Delta\mathcal{M}_t^i\right\|^2\right] = \mathrm{E}\left[\sum_{i=1}^{N} \frac{1}{p_i} \left\|\Delta\mathcal{M}_t^i\right\|^2\right] - \mathrm{E}\left[\sum_{i=1}^{N} \left\|\Delta\mathcal{M}_t^i\right\|^2\right]$$

The optimal probabilities are obtained by minimizing α_C w.r.t. $\{p_i\}_{i=1}^N$ subject to the constraints $0 \leq p_i \leq 1$ and $m \geq b = \sum_{i=1}^N p_i$.

Proof. This proof uses an argument similar to that in the proof of Lemma 2 in [1] (Horváth & Richtárik, 2019). The Lagrangian of our optimization problem is given by:

$$L\left(\{p_i\}_{i=1}^N, \{\lambda_i\}_{i=1}^N, \{u_i\}_{i=1}^N, y\right) = \alpha_C\left(\{p_i\}_{i=1}^N\right) - \sum_{i=1}^{N} \lambda_i p_i - \sum_{i=1}^{N} u_i\left(1-p_i\right) - y\left(m - \sum_{i=1}^{N} p_i\right).$$

Since all constraints are linear and the support of $\{p_i\}_{i=1}^N$ is convex, the KKT conditions hold. Therefore, the following solution is deduced from the KKT conditions:

$$p_i = \begin{cases} \frac{(m+l-N)*\left\|\Delta\mathcal{M}_t^i\right\|}{\sum_{j=1}^{l}\left\|\Delta\mathcal{M}_t^{(j)}\right\|}, & \text{if } \left\|\Delta\mathcal{M}_t^i\right\| < \left\|\Delta\mathcal{M}_t^{(l+1)}\right\| \\ 1, & \text{otherwise} \end{cases}$$

where $\left\|\Delta\mathcal{M}_t^{(j)}\right\|$ is the j-th largest value among the values $\left\|\Delta\mathcal{M}_t^1\right\|$,$\left\|\Delta\mathcal{M}_t^2\right\|$, ...,$\left\|\Delta\mathcal{M}_t^N\right\|$; l is the largest integer for which $0 < m + l - N \leq \frac{\sum_{i=1}^{l}\left\|\Delta\mathcal{M}_t^{(i)}\right\|}{\left\|\Delta\mathcal{M}_t^{(l)}\right\|}$.

Proof 4 - Optimal Sampling selects Attacked Clients. In the context of federated learning aimed at training a binary classifier, with malicious client e_m and a random benign client e_b. We consider the case when e_m and e_b has benign data of same distribution (denote $\mathcal{D}_{clean}$ as dataset of this distribution), while e_m has additionally a small set of backdoor data (denote $\mathcal{D}_{backdoor}$ with $|\mathcal{D}_{backdoor}| = \xi * |\mathcal{D}_{clean}|$).

Let $x, \tilde{x}$ be samples from $\mathcal{D}_{clean}$ (label '0') and $\mathcal{D}_{backdoor}$ (label '1'), respectively. At round t, global model $\mathcal{M}$ is sent to e_b and e_m. Define F as function that plays a role as Feature Extractor and W is penultimate layer of $\mathcal{M}$, i.e. $\mathcal{M}(.) = softmax(W * F(.))$. The following condition can assure the selection on attacked clients:

Lemma 2. *For any function $F = [F^1; F^2; ...; F^d] : \mathbb{R}^s \to \mathbb{R}^d_{\geq 0}$ such that each function F^i is twice-differentiable and has continuous derivatives in an open ball B with radius $\Delta x = \tilde{x} - x$ around the point x. If Hessian Matrix of each function F^i is semi-positive definite at any points between x and $\tilde{x}$, and this condition satisfies $\forall i$:*

$$\Delta x^T \nabla F^i(x) > \frac{2}{\xi} * F^i(x) \tag{3}$$

then

$$p_m > p_b \tag{4}$$

where p_m, p_b are probabilities for saving client e_m and e_b, respectively.

Proof. We have that $\forall i : F^i$ is twice-differentiable and has continuous derivatives in an open ball of radius Δx. Implement Multivariate Taylor's expansion for F^i around point x, note that $\tilde{x} = x + \Delta x$:

$$F^i(\tilde{x}) = F^i(x) + \Delta x^T \nabla F^i(x) + \frac{1}{2}(\Delta x)^T(\nabla^2 F^i(x_0))(\Delta x) \tag{5}$$

where x_0 is a point that lies between x and $\tilde{x}$ and $\nabla^2 f(x_0)$ is the Hessian of f evaluated at a point x_0.
Because Hessian Matrix of F^i is semi-positive, $\frac{1}{2}(\Delta x)^T(\nabla^2 F^i(x_0))(\Delta x) > 0$. Combine with the condition (3), we have $\forall i$:

$$F^i(\tilde{x}) > \frac{\xi + 2}{\xi} F^i(x) \tag{6}$$

Back to our analysis on gradient, we first compute gradient on W regards to x and $\tilde{x}$.

Update in one cell of W:
For benign client e_b:

$$\Delta w_{rc} = -\eta \mathrm{E}\left[\frac{\partial \mathcal{L}(W, x; y_r)}{\partial w_{rc}}\right]$$

For malicious client e_m:

$$\begin{aligned}\Delta w_{rc} = -\eta(& \frac{|\mathcal{D}_{clean}|}{|\mathcal{D}_{clean}| + |\mathcal{D}_{backdoor}|}\mathrm{E}\left[\frac{\partial \mathcal{L}(W, x; y_r)}{\partial w_{rc}}\right] \\ & + \frac{|\mathcal{D}_{backdoor}|}{|\mathcal{D}_{clean}| + |\mathcal{D}_{backdoor}|}\mathrm{E}\left[\frac{\partial \mathcal{L}(W, \tilde{x}; y_r)}{\partial w_{rc}}\right]) \\ = & -\eta\frac{1}{1+\xi}(\mathrm{E}\left[\frac{\partial \mathcal{L}(W, x; y_r)}{\partial w_{rc}}\right] + \xi \mathrm{E}\left[\frac{\partial \mathcal{L}(W, \tilde{x}; y_r)}{\partial w_{rc}}\right])\end{aligned}$$

Note that:

$$\frac{\partial \mathcal{L}(W, x; y_r)}{\partial w_{rc}} = (softmax(W * F(x))_r - y_r) * F^c(x)$$

and

$$\sum_r L(x)_{rc} = F^c(x) \sum_r (softmax(WF(x))_r - y_r) = 0$$

where $L(x)_{rc} = \partial \mathcal{L}(W, x; y_r)/\partial w_{rc}$.

Then square of L2-norm for updates on e_b and e_m are respectively shown as:

$$||\Delta_b W||^2 = 2\eta^2 \sum_c (\mathrm{E}\left[L(x)_{0c}\right])^2$$

and

$$||\Delta_m W||^2 = \frac{2\eta^2}{(1+\xi)^2} \sum_c (\mathrm{E}\left[L(x)_{0c}\right] + \xi \mathrm{E}\left[L(\tilde{x})_{0c}\right])^2$$

From that, we have:

$$\begin{aligned}||\Delta_m W||^2 - ||\Delta_b W||^2 = & \frac{2\eta^2\xi}{(1+\xi)^2} \sum_c (\mathrm{E}\left[L(\tilde{x})_{0c}\right] \\ & - \mathrm{E}\left[L(x)_{0c}\right])(\xi \mathrm{E}\left[L(\tilde{x})_{0c}\right] + (\xi + 2)\mathrm{E}\left[L(x)_{0c}\right])\end{aligned} \tag{7}$$

Since x has label '0' and $\tilde{x}$ has label '1', $\mathrm{E}\left[L(x)_{0c}\right] = (softmax(W * F(x))_0 - 1) * F^c(x) < 0$ and $\mathrm{E}\left[L(\tilde{x})_{0c}\right] = softmax(W * F(\tilde{x}))_0 * F^c(x) > 0$. Moreover, we have $softmax(W * F(\tilde{x}))_0 > \mu > softmax(W * F(x))_1$, so:

$$\xi \mathrm{E}\left[L(\tilde{x})_{0c}\right] + (\xi + 2)\mathrm{E}\left[L(x)_{0c}\right] > \mu(\xi \mathrm{E}\left[F^c(\tilde{x})\right] - (\xi + 2)\mathrm{E}\left[F^c(x)\right]) \quad (8)$$

Due to (6), $\xi \mathrm{E}\left[L(\tilde{x})_{0c}\right] + (\xi + 2)\mathrm{E}\left[L(x)_{0c}\right] > 0$. Hence, we have $||\Delta_m W|| > ||\Delta_b W||$.

We consider these cases of e_m and e_b:

1. $m, b \in A^k$ or $m, b \notin A^k$, easily to see that $p_m > p_b$.
2. $m \in A^k$ and $b \notin A^k$, $p_m = 1 > p_b$.
3. $m \notin A^k$ and $b \in A^k$, then $||\Delta_b W|| \geq ||\Delta W_{(l+1)}|| > ||\Delta W_{(l)}|| \geq ||\Delta_m W||$. (absurd)

In all cases, we have $p_m > p_b$.

References

1. Horváth, S., Richtárik, P.: Nonconvex variance reduced optimization with arbitrary sampling. In: ICLR. pp. 2781–2789 (2019)