Appendix for the Paper "Integrating Independent Layer-Wise Rank Selection with Low-Rank SVD Training for Model Compression: A Theory-Driven Approach"

A Proof of Theorems 1, 2, and 3

Proposition 1 (Theorem 4.2 [Wright and Ma, 2022]). *Let $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ be a matrix, and $r = \mathrm{rank}(\boldsymbol{W})$. $\boldsymbol{W}$ can be decomposed as $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$, such that $\boldsymbol{U}\boldsymbol{U}^T = \mathbf{I}$ and $\boldsymbol{V}\boldsymbol{V}^T = \mathbf{I}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$ is a is diagonal matrix, i.e., $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma)$, $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_r]$, and $\sigma_k (k \in [r])$ are singular values of $\boldsymbol{W}$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. Then, we have $\boldsymbol{W} = \sum_{i=1}^r \sigma_i \boldsymbol{U}_{:,i}\boldsymbol{V}_{i,:}^T$.*

Lemma 1 (Eckart–Young–Mirsky Theorem [Golub *et al.*, 1987]). *Let $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ be a matrix, and $r = \mathrm{rank}(\boldsymbol{W})$. Following the same settings in Proposition 1, we define $\boldsymbol{W}_k$ to be the best rank-k approximation to $\boldsymbol{W}$ in the spectral norm, i.e., $\boldsymbol{W}_k = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T = \sum_{i=1}^k \sigma_i \boldsymbol{U}_{:,i}\boldsymbol{V}_{i,:}^T$, where $\boldsymbol{U}_k, \boldsymbol{\Sigma}_k, \boldsymbol{V}_k$ are top-k vectors truncated from $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}$. Then, we have $||\boldsymbol{W} - \boldsymbol{W}_k||_2 = \sigma_{k+1}$, where $||\cdot||_2$ stands for the spectral norm.*

Proof. We have

$$\begin{aligned}
||\boldsymbol{W} - \boldsymbol{W}_k||_2 &= ||\sum_{i=1}^r \sigma_i \boldsymbol{V}_{i,:}\boldsymbol{U}_{:,i}^T - \sum_{i=1}^k \sigma_i \boldsymbol{V}_{i,:}\boldsymbol{U}_{:,i}^T||_2 && \text{(Proposition 1)}\\
&= ||\sum_{i=k+1}^r \sigma_i \boldsymbol{V}_{i,:}\boldsymbol{U}_{:,i}^T||_2\\
&= \sigma_{k+1} && \text{(The definition of spectral norm)}
\end{aligned}$$

□

Theorem 1 (The output difference bound for rank-k approximation over L-layer neural networks). *We denote a^l to be the activation function for the l-th layer, and assume a^l is ρ_l-Lipschitz and $a^l(0) = 0$ for all $l \in [1, L]$. Let X^0 be the initial input vector, X^l and X_k^l be the output vectors as a result of passing the full-rank matrix $\boldsymbol{W}^l$ and low-rank matrix $\boldsymbol{W}_k^l$ through the l-th layer, respectively, and σ_i^l be the i-th singular value of $\boldsymbol{W}^l$. We define k^l such that the top k^l largest singular values of the full-rank matrix $\boldsymbol{W}^l$ are kept in the corresponding low-rank SVD approximated matrix $\boldsymbol{W}_k^l$ in layer l. Then, the output difference from rank-k approximation over L-layer feed-forward networks $||X^L - X_k^L||_2$ is upper-bounded by $\left(\prod_{l=1}^L \rho_l\sigma_1^l\right)\left(\sum_{l=1}^L \frac{\sigma_{k^l+1}^l}{\sigma_1^l}\right)||X^0||_2$.*

Proof. For the output difference at layer $l+1$, we have

$$\begin{aligned}
||X^{l+1} - X_k^{l+1}||_2 &= ||a^{l+1}(\boldsymbol{W}^{l+1}X^l) - a^{l+1}(\boldsymbol{W}_k^{l+1}X_k^l)||_2\\
&\leq \rho_{l+1}||\boldsymbol{W}^{l+1}X^l - \boldsymbol{W}_k^{l+1}X_k^l||_2\\
&\leq \rho_{l+1}||\boldsymbol{W}^{l+1}X^l - \boldsymbol{W}_k^{l+1}X^l + \boldsymbol{W}_k^{l+1}X^l - \boldsymbol{W}_k^{l+1}X_k^l||_2\\
&\leq \rho_{l+1}||\boldsymbol{W}^{l+1}X^l - \boldsymbol{W}_k^{l+1}X^l||_2 + \rho_{l+1}||_2\boldsymbol{W}_k^{l+1}X^l - \boldsymbol{W}_k^{l+1}X_k^l||_2\\
&\leq \rho_{l+1}||\boldsymbol{W}^{l+1} - \boldsymbol{W}_k^{l+1}||_2 \cdot ||X^l||_2 + \rho_{l+1}||\boldsymbol{W}_k^{l+1}||_2 \cdot ||X^l - X_k^l||_2\\
&\leq \rho_{l+1}\sigma_{k^{l+1}+1}^{l+1}||X^l||_2 + \rho_{l+1}\sigma_1^{l+1} \cdot ||X^l - X_k^l||_2. && \text{(Lemma 1)}
\end{aligned}$$

Also, we notice that,

$$||X^l||_2 \leq \left(\prod_{i=1}^l \rho_i ||\boldsymbol{W}^i||_2\right)||X^0||_2 = \left(\prod_{i=1}^l \rho_i\sigma_1^i\right)||X^0||_2.$$

If we let $c_{l+1} = \rho_{l+1}\sigma_{k^{l+1}+1}^{l+1}\left(\prod_{i=1}^l \rho_i\sigma_1^i\right)$ and $d_{l+1} = \rho_{l+1}\sigma_1^{l+1}$, then we have,

$$||X^{l+1} - X_k^{l+1}||_2 \leq c_{l+1}||X^0||_2 + d_{l+1}||X^l - X_k^l||_2.$$

By induction over L layers, we have,

$$\begin{aligned}
||X^L - X_k^L||_2 &\leq c_L||X^0||_2 + d_L||X^{L-1} - X_k^{L-1}||_2 \\
&\leq c_L||X^0||_2 + d_L\left(c_{L-1}||X^0||_2 + d_{L-1}||X^{L-2} - X_k^{L-2}||_2\right) \\
&= \left(c_L + d_Lc_{L-1}\right)||X^0||_2 + d_Ld_{L-1}||X^{L-2} - X_k^{L-2}||_2 \\
&\leq \left(c_L + d_Lc_{L-1} + d_Ld_{L-1}c_{L-2}\right)||X^0||_2 + d_Ld_{L-1}d_{L-2}||X^{L-3} - X_k^{L-3}||_2 \\
&\dots \\
&\leq \left(c_L + d_Lc_{L-1} + d_Ld_{L-1}c_{L-2} + \cdots + \left(\prod_{l=0}^{L-2} d_{L-l}\right)c_1\right)||X^0||_2 + \left(\prod_{l=1}^{L-1} d_{L-l}\right)||X^0 - X^0||_2 \\
&= \left(c_L + d_Lc_{L-1} + d_Ld_{L-1}c_{L-2} + \cdots + \left(\prod_{l=0}^{L-2} d_{L-l}\right)c_1\right)||X^0||_2 \\
&= \left(\prod_{l=1}^{L} \rho_l\sigma_1^l\right)\left(\sum_{l=1}^{L} \frac{\sigma_{k^l+1}^l}{\sigma_1^l}\right)||X^0||_2
\end{aligned}$$

□

Theorem 2 (The loss error bound from rank-k approximation in classification problems)**.** *Following the settings in Theorem 1, we consider a C-class classification problem. Let $X_i^L \in \mathbb{R}^C$ and $X_{i,k}^L \in \mathbb{R}^C$ be the output logits when feeding a input X_i^0 sampled from the training dataset $\mathcal{D}^{tr}$, e.g., $\mathcal{D}^{tr} = \{X_i^0, y_i\}_{i=1}^R$, from the full-rank parameter space $\boldsymbol{\mathcal{W}}$ and low-rank parameter space $\boldsymbol{\mathcal{W}_k}$, respectively. Particularly, $\boldsymbol{\mathcal{W}} = \{\boldsymbol{W}^1, \boldsymbol{W}^2, \dots, \boldsymbol{W}^L\}$, $\boldsymbol{\mathcal{W}_k} = \{\boldsymbol{W}_k^1, \boldsymbol{W}_k^2, \dots, \boldsymbol{W}_k^L\}$, $X_i^L = f_{\boldsymbol{\mathcal{W}}}(X_i^0)$, $X_{i,k}^L = f_{\boldsymbol{\mathcal{W}_k}}(X_{i,k}^0)$, and $||X_i^0||_2 \leq B$, for $\forall\ i \in [1, R]$. Let $z_i = \texttt{softmax}(X_i^L)$ and $z_{i,k} = \texttt{softmax}(X_{i,k}^L)$, where `softmax` is the softmax function. We consider the cross-entropy function as the loss function, i.e., $g(z, y) = -y^T\log(z)$. Let $L(\boldsymbol{\mathcal{W}}; X_i^0) = g(z_i, y_i)$ and $L(\boldsymbol{\mathcal{W}_k}; X_i^0) = g(z_{i,k}, y_i)$. Now, we set $\frac{\sigma_{k^l+1}^l}{\sigma_1^l} < \delta$, $\forall\ l \in [1, L]$. Then, we have, for $\forall\ \epsilon > 0$, $\exists\ \delta = \frac{\epsilon}{\sqrt{2}BL(\prod_{l=1}^L \rho_l\sigma_1^l)}$, s.t. $|L(\boldsymbol{\mathcal{W}}; \mathcal{D}^{tr}) - L(\boldsymbol{\mathcal{W}_k}; \mathcal{D}^{tr})| < \epsilon$.*

Proof. First, we define $\frac{\partial L(\boldsymbol{\mathcal{W}}, X_i^0)}{\partial x} \in \mathbb{R}^C$ to be the partial derivative of $L(\boldsymbol{\mathcal{W}}, X_i^0)$ with respect to the output layer X_k^L, for the given input X_i^0. Then, we have

$$\left|\left|\frac{\partial L(\boldsymbol{\mathcal{W}}, X_i^0)}{\partial x}\right|\right|_2 = ||z_i - y_i||_2.$$

Without loss of generality, for $z_i = [z_{i,1}, \dots, z_{i,C}] \in \mathbb{R}^C, y_i = [y_{i,1}, \dots, y_{i,C}] \in \mathbb{R}^C$, we assume $y_{i,c} = 0$, where $c \in [1, C-1]$ and $y_{1,C} = 1$ and $z_{i,1} + \cdots + z_{i,C} = 1$, where $z_{i,j} \in [0, 1], \forall\ j \in [1, C]$, Then

$$\begin{aligned}
\left|\left|\frac{\partial L(\boldsymbol{\mathcal{W}}, X_i^0)}{\partial x}\right|\right|_2 &= ||z_i - y_i||_2 \\
&= \sqrt{(z_{i,1} - y_{i,1})^2 + (z_{i,2} - y_{i,2})^2 + \cdots + (z_{i,C} - y_{i,C})^2} \\
&= \sqrt{z_{i,1}^2 + z_{i,2}^2 + \cdots + z_{i,C-1}^2 + (z_{i,C} - 1)^2} \\
&= \sqrt{z_{i,1}^2 + \cdots + z_{i,C-1}^2 + z_{i,C}^2 - 2z_{i,C} + 1} \\
&= \sqrt{z_{i,1}^2 + \cdots + z_{i,C-1}^2 + z_{i,C}^2 - 2(z_{i,1} + \cdots + z_{i,C})z_{i,C} + 1} \\
&= \sqrt{(z_{i,1} - z_{i,C})^2 + \cdots + (z_{i,C-1} - z_{i,C})^2 - Cz_{i,C}^2 + 1} \\
&\leq \sqrt{(z_{i,1} + \cdots + z_{i,C-1} - (C-1)z_{i,C})^2 - Cz_{i,C}^2 + 1} \\
&\quad \text{(The equality case holds when } (z_{i,j} - z_{i,C})(z_{i,j'} - z_{i,C}) = 0 \text{ for } j, j' \in [1, C-1]) \\
&= \sqrt{(1 - Cz_{i,C})^2 - Cz_{i,C}^2 + 1} \\
&\leq \sqrt{2} \qquad \text{(The equality case holds when } z_{i,C} = 0)
\end{aligned}$$

Overall, the equality case occurs when $z_{i,j} = 1$ if $j \neq C$ and all rest $z_{i,j'}$ are 0 if $j' \neq j$ and $j' \in [1, C]$.

Let $\delta = \frac{\epsilon}{\sqrt{2}BL(\prod_{l=1}^{L}\rho_l\sigma_1^l)}$, where $||X_i^0||_2 \leq B$ and $\frac{\sigma_{k^l+1}^l}{\sigma_1^l} < \delta, \forall\, l \in [L]$. With Lagrange's mean value theorem, we have

$$\begin{aligned}
|L(\mathcal{W}; X_i^0) - L(\mathcal{W}_{\boldsymbol{k}}; X_i^0)| &\leq \max_{\substack{x \in tX_k^L + (1-t)X^L \\ t\in[0,1]}} \left\{ \left|\left| \frac{\partial L}{\partial x} \right|\right|_2 \right\} ||X^L - X_k^L||_2 \\
&\leq \sqrt{2}||X^L - X_k^L||_2 \\
&\leq \sqrt{2}||X_i^0||_2 \sum_{l=1}^{L} \frac{\sigma_{k^l+1}^l}{\sigma_1^l} \left(\prod_{l=1}^{L} \rho_l \sigma_1^l \right) \\
&< \sqrt{2}B\delta L \left(\prod_{l=1}^{L} \rho_l \sigma_1^l \right) = \epsilon.
\end{aligned}$$

Finally, $|L(\mathcal{W}; \mathcal{D}^{tr}) - L(\mathcal{W}_{\boldsymbol{k}}; \mathcal{D}^{tr})| \leq \frac{1}{R}\sum_{i=1}^{R} |L(\mathcal{W}; X_i^0) - L(\mathcal{W}_{\boldsymbol{k}}; X_i^0)| < \epsilon$. □

Theorem 3 (The loss error bound from rank-k approximation in regression problems)**.** *Following the settings in Theorem 1, we consider a regression problem. Let X_i^L and $X_{i,k}^L$ be the output at layer L when feeding a input X_i^0 sampled from the training dataset $\mathcal{D}^{tr}$, e.g., $\mathcal{D}^{tr} = \{X_i^0, y_i\}_{i=1}^R$, from the full-rank parameter space $\mathcal{W}$ and low-rank parameter space $\mathcal{W}_{\boldsymbol{k}}$, respectively. Particularly, $\mathcal{W} = \{\boldsymbol{W}^1, \boldsymbol{W}^2, \ldots, \boldsymbol{W}^L\}$, $\mathcal{W}_{\boldsymbol{k}} = \{\boldsymbol{W}_k^1, \boldsymbol{W}_k^2, \ldots, \boldsymbol{W}_k^L\}$, $X_i^L = f_{\mathcal{W}}(X_i^0)$, $X_{i,k}^L = f_{\mathcal{W}_{\boldsymbol{k}}}(X_{i,k}^0)$, and $||X_i^0||_2 \leq B$, for $\forall\, i \in [1, R]$. We consider the loss function as $g(z, y) = ||z - y||_2$. Let $L(\mathcal{W}; X_i^0) = g(X_i^L, y_i)$ and $L(\mathcal{W}_{\boldsymbol{k}}; X_i^0) = g(X_{i,k}^L, y_i)$. Now, we set $\frac{\sigma_{k^l+1}^l}{\sigma_1^l} < \delta$, $\forall\, l \in [1, L]$. Then, we have, for $\forall\, \epsilon > 0$, $\exists\, \delta = \frac{\epsilon}{BL(\prod_{l=1}^{L}\rho_l\sigma_1^l)}$, s.t. $|L(\mathcal{W}; \mathcal{D}^{tr}) - L(\mathcal{W}_{\boldsymbol{k}}; \mathcal{D}^{tr})| < \epsilon$.*

Proof. Note that

$$\begin{aligned}
|L(\mathcal{W}; X_i^0) - L(\mathcal{W}_{\boldsymbol{k}}; X_i^0)| &= \big|\, ||X_i^L - y_i||_2 - ||X_{i,k}^L - y_i||_2 \,\big| \\
&\leq ||(X_i^L - y_i) - (X_{i,k}^L - y_i)||_2 \\
&= ||X_i^L - X_{i,k}^L||_2.
\end{aligned}$$

Let $\delta = \frac{\epsilon}{BL\sqrt{2}(\prod_{l=1}^{L}\rho_l\sigma_1^l)}$, where $||X_i^0||_2 \leq B$ and $\frac{\sigma_{k^l+1}^l}{\sigma_1^l} < \delta, \forall\, l \in [L]$. We have

$$\begin{aligned}
|L(\mathcal{W}; X_i^0) - L(\mathcal{W}_{\boldsymbol{k}}; X_i^0)| &\leq ||X^L - X_k^L||_2 \\
&\leq ||X_i^0||_2 \sum_{l=1}^{L} \frac{\sigma_{k^l+1}^l}{\sigma_1^l} \left(\prod_{l=1}^{L} \rho_l \sigma_1^l \right) \\
&< B\delta L \left(\prod_{l=1}^{L} \rho_l \sigma_1^l \right) = \epsilon.
\end{aligned}$$

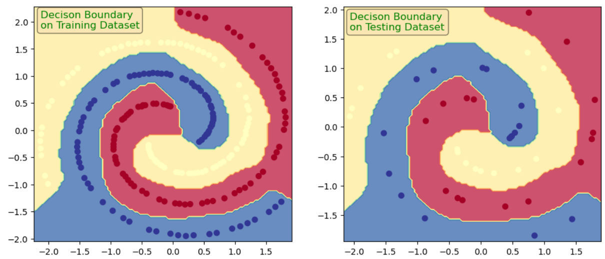
Finally, $|L(\mathcal{W}; \mathcal{D}^{tr}) - L(\mathcal{W}_{\boldsymbol{k}}; \mathcal{D}^{tr})| \leq \frac{1}{R}\sum_{i=1}^{R} |L(\mathcal{W}; X_i^0) - L(\mathcal{W}_{\boldsymbol{k}}; X_i^0)| < \epsilon$. □

B Statistical Analysis of Our Pilot Study in Section 4.1

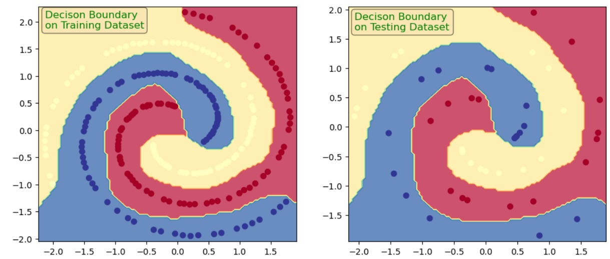
	Full-Rank	Low-Rank				
ϵ	0	0.17	0.23	0.28	0.33	0.56
δ	0	0.015	0.021	0.025	0.03	0.047
k^1 (Layer 1)	2	2	2	2	2	2
k^2 (Layer 2)	100	9	8	7	6	3
k^3 (Layer 3)	3	3	3	3	3	3

Table 4: Statistics of our pilot study.

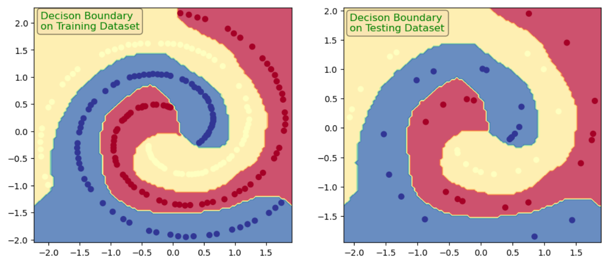
To validate the feasibility of identifying δ based on our derived ϵ-δ correlation and determining the optimal k^l, we conduct a pilot study using a simple 3-layer feed-forward neural network for a ternary classification problem. Particularly, the input is in 2 dimensions; the output is in 3 dimensions; $\boldsymbol{W}^1 \in \mathbb{R}^{2\times100}$, $\boldsymbol{W}^2 \in \mathbb{R}^{100\times100}$, and $\boldsymbol{W}^1 \in \mathbb{R}^{100\times3}$. Here, $\boldsymbol{W}^1$, $\boldsymbol{W}^2$,



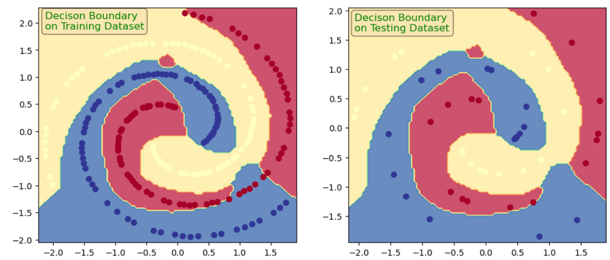
(a) Full-rank model space ($k^2 = 100$)



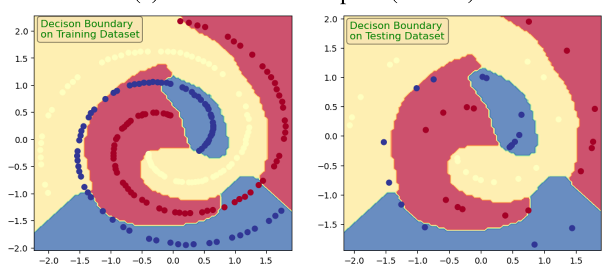
(b) Low-rank model space ($k^2 = 9$)



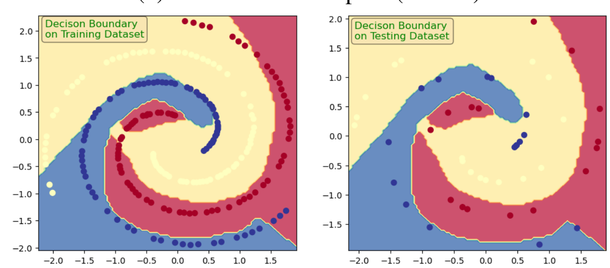
(c) Low-rank model space ($k^2 = 8$)



(d) Low-rank model space ($k^2 = 7$)



(e) Low-rank model space ($k^2 = 6$)



(f) Low-rank model space ($k^2 = 3$)

Figure 7: A visualization of the decision boundaries on the training dataset (left column) and testing dataset (right column) through different rank selections.

and $\boldsymbol{W}^3$ are either column full-rank or row full-rank. It means that $\mathrm{rank}(\boldsymbol{W}^1) = 2$, $\mathrm{rank}(\boldsymbol{W}^2) = 100$, and $\mathrm{rank}(\boldsymbol{W}^3) = 3$. Since there is not much room to tune k in $\boldsymbol{W}^1$ and $\boldsymbol{W}^3$, our major focus is on tuning k in $\boldsymbol{W}^2$. The activation functions in the first and second layers are both the ReLU functions. The output from the third layer will be fed into a softmax function to get the prediction probabilities. The loss function is the cross-entropy function. Fig. 7 visualizes the decision boundary on the training (left column in each sub-figure) and testing dataset (right column in each sub-figure) through different rank selections with more details. Table 4 lists the ϵ, δ, and selected k for each layer. As we can find, as the loss error bound ϵ increases, the corresponding δ also increases, indicating that more information is truncated as smaller values of k^l are assigned to each layer, respectively. This truncation leads to underfitting in the low-rank model, preventing it from effectively capturing the patterns of the original full-rank model. Particularly, according to the results in Fig. 7, the key turning point in selecting k^2 happens from 8 to 7. We can find the decision boundary's shape has appeared different patterns compared with that in larger k^2. To some extent, when $k^2 = 8$, it represents the smallest k^l, which is sufficient to retain the full-rank model's representational capacity. Further searching would significantly degrade the performance.

C Our Proposed Rank Selection Enabled Low-Rank SVD Training Algorithm

We present our proposed rank selection enabled low-rank SVD training algorithm in detail, as illustrated in Algorithms 3, 4, and 5.

Algorithm 3: Proposed Rank Selection Enabled Low-Rank SVD Training Algorithm

Input: full-rank parameters $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\mathcal{V}}$, loss error tolerance ϵ, stop searching precision $\Delta\delta$, training epoch E
Output: low-rank parameters $\boldsymbol{\mathcal{U}}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mathcal{V}}_k$

Initialize parameters $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\mathcal{V}}$;
$e \leftarrow 1$;
while $e \leq E$ **do**
    Update $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\mathcal{V}}$ based on loss function $L(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}})$ with an appropriate optimizer and extract the learning loss $L_T(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}})$; `//` $L_O(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{V}})$ `and` $L_R(\boldsymbol{\Sigma})$ `are not used in the next-step rank selection`
    $\boldsymbol{\mathcal{U}}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mathcal{V}}_k \leftarrow$ `RankSelection`$(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}}, \epsilon, \Delta\delta, L_T(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}}))$;
    $\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}} \leftarrow \boldsymbol{\mathcal{U}}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mathcal{V}}_k$; `// Use truncated models for the next round of training`
    $e \leftarrow e + 1$;
return $\boldsymbol{\mathcal{U}}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mathcal{V}}_k$;

Algorithm 4: Rank Selection Function

Function `RankSelection` ($\boldsymbol{\mathcal{U}}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\mathcal{V}}$, ϵ, $\Delta\delta$, $L_T(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}})$) **:**
    $floss \leftarrow L_T(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}})$;
    $l \leftarrow 0$; $u \leftarrow 1$; $\delta \leftarrow (l + u)/2$;
    while $|l - u| \geq \Delta\delta$ *or* $|floss - loss| \geq \epsilon$ **do**
        $\boldsymbol{\mathcal{U}}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mathcal{V}}_k \leftarrow$ `Truncation`$(\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}}, \delta)$;
        $loss \leftarrow L_T(\boldsymbol{\mathcal{U}}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mathcal{V}}_k)$;
        if $|floss - loss| < \epsilon$ **then**
            $l \leftarrow \delta$; $\delta \leftarrow (l + u)/2$;
        else
            $u \leftarrow \delta$; $\delta \leftarrow (l + u)/2$;
    return $\boldsymbol{\mathcal{U}}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mathcal{V}}_k$;

Algorithm 5: Truncation Function

Function `Truncation` ($\boldsymbol{\mathcal{U}}, \boldsymbol{\Sigma}, \boldsymbol{\mathcal{V}}, \delta$) **:**
    $\boldsymbol{\mathcal{U}}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\mathcal{V}}_k \leftarrow \emptyset$;
    for *each* $\boldsymbol{U}^l \in \boldsymbol{\mathcal{U}}$, $\boldsymbol{\Sigma}^l \in \boldsymbol{\Sigma}$, $\boldsymbol{V}^l \in \boldsymbol{\mathcal{V}}$ **do**
        $k^l \leftarrow \text{argmax}_k\{k | \sigma_k^l / \sigma_1^l \geq \delta\}$;
        Truncate top-k^l vectors from $\boldsymbol{U}^l, \boldsymbol{\Sigma}^l, \boldsymbol{V}^l$ to get $\boldsymbol{U}_k^l, \boldsymbol{\Sigma}_k^l, \boldsymbol{V}_k^l$, and add into $\boldsymbol{\mathcal{U}}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mathcal{V}}_k$;
    return $\boldsymbol{\mathcal{U}}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mathcal{V}}_k$