Supplementary Material for:
"Sparsity Outperforms Low-Rank Projections in Few-Shot Adaptation"

Nairouz Mrabah[1] Nicolas Richet[1] Ismail Ben Ayed[1] Eric Granger[1]
[1]LIVIA, ILLS, Department of Systems Engineering, ÉTS Montreal, Québec, Canada
mrabah.nairouz@livia.etsmtl.ca, nicolas.richet.1@ens.etsmtl.ca,
ismail.benayed@etsmtl.ca, eric.granger@etsmtl.ca

A. Few-shot Adaptation of VLMs

VLMs are designed to learn a joint representation space for images and text. They are typically trained on large-scale image-text pairs. By aligning visual and textual information, VLMs can perform tasks such as zero-shot classification and cross-modal retrieval. CLIP is a type of VLM that relies on contrastive learning to associate images with their corresponding textual descriptions in the latent space. Given a batch of N image-text pairs $\{(x_i, t_i)\}_{i=1}^N$, where x_i represents an image and t_i its associated text, CLIP maximizes the similarity between matching pairs while minimizing it for non-matching ones.

Formally, let $f_\theta(\cdot)$ and $g_\phi(\cdot)$ be the vision and text encoders, respectively, where θ and ϕ denote their learnable parameters. Given an image x_i and its corresponding text t_i, the encoders project them into a shared normalized embedding space. The image embedding is given by $v_i = f_\theta(x_i)$, and the text embedding is given by $z_i = g_\phi(t_i)$. Both embeddings lie in the joint representation space. The VLM model is then trained using a symmetric InfoNCE loss.

Zero-shot Inference. Once pretrained, CLIP classifies images without additional training on the target task. Classification is performed by comparing the image embedding to predefined text embeddings representing class labels. For a classification task with C categories, each class c is associated with a set of n text prompts $\{t_{j,c}\}_{j=1}^n$. The prototype for class c is computed by averaging the embeddings of its prompts $z_c = \frac{1}{n}\sum_{j=1}^n g_\phi(t_{j,c})$. The model predicts the class using the softmax over the cosine similarities between the image embedding v_i and the class prototypes z_c:

$$\hat{y}_{ic} = \frac{\exp\left((v_i^\top \cdot z_c)/\tau\right)}{\sum_{j=1}^C \exp\left((v_i^\top \cdot z_j)/\tau\right)}, \tag{1}$$

where $\hat{y}_{ic}$ represents the probability of the image x_i belonging to class c, and τ is a temperature parameter that scales the logits to control the sharpness of the probability distribution. Since the embeddings v_i and z_c are ℓ_2-normalized, the cosine similarity simplifies to the dot product operation.

Few-shot Learning. Few-shot learning addresses the challenge of adapting VLMs to new tasks using only a limited number of labeled examples per class. Formally, let $S = \{(x_i, y_i)\}_{i=1}^{K\times C}$ be the support set, where K is the number of examples per class. K typically takes small values, such as $K \in \{1, 2, 4\}$. Each label $y \in \{0,1\}^C$ is represented as a one-hot vector, where only one dimension is active to indicate the correct class. The objective is to adapt the pretrained VLM efficiently by leveraging this small support set S while preserving its generalization ability.

To optimize the VLM model under the few-shot setting, the cross-entropy loss function is typically used. Given the support set S, the objective is to minimize:

$$\mathcal{L}_{CE} = -\frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{C} y_{ij}\,\ln \hat{y}_{ij}. \tag{2}$$

This loss function maximizes the likelihood of the correct class labels and minimizes incorrect predictions.

B. Algorithm

Our optimization strategy is described in Algorithm 1. By promoting random gradient selection and importance-based moment pruning, SO balances the short-term updates with the long-term significance of parameters.

C. Memory Consumption and Scalability

Memory consumption. able 1 compares SO to various low-rank methods and Adam in theoretical memory usage for a single linear layer ($W \in \mathbb{R}^{m\times n}$). It illustrates how each method's weight, gradient, and optimizer states scale w.r.t. rank r or sparsity ratio $1-\kappa$. Notably, SO stores $2mn\kappa$ and $3mn\kappa$ parameters for the gradient and optimizer states, respectively, offering significant savings when κ is small.

Table 2 extends this analysis to the full CLIP model, covering all 12 blocks of both the text and vision encoders. We exclude biases and activations since they are shared across all approaches. Our experiments use a default sparsity ratio

Table 1. Comparison of SO, GaLoRE, LoRA, PiSSA, DoRA, ReLoRA, VeRA, and Adam in memory requirements for a single fully-connected layer. Denote the weight of the layer $W \in \mathbb{R}^{m \times n}$, n the input dimension, m the output dimension, $(m \leq n)$, rank r, and sparsity ratio $1 - \kappa$.

	SO	GaLoRE	LoRA	PiSSA	DoRA	ReLoRA	VeRA	Adam
Weight	mn	mn	$mn + mr + nr$	$mn + mr + nr$	$mn + mr + nr + m$	$mn + mr + nr$	$mn + mr + nr + m + r$	mn
Gradient	$2mn\kappa$	mn	$mr + nr$	$mr + nr$	$mr + nr + m$	$mr + nr$	$m + r$	mn
Optimizer States	$3mn\kappa$	$mr + 2nr$	$2mr + 2nr$	$2mr + 2nr$	$2mr + 2nr + 2m$	$2mr + 2nr$	$2m + 2r$	$2mn$

Table 2. Comparison of theoretical memory consumption for CLIP when adapting all vision and text transformer blocks. Biases and activations are excluded since they are shared across all methods. The table reports the total number of variables, overall memory usage, and trainable parameters.

Method	Weight (#Vars, MB)	Gradient (#Vars, MB)	Opt. States (#Vars, MB)	#Trainable	Total Mem. (MB)
SO ($\kappa = 0.05\%$)	122683392 (468MB)	122683 (0.47MB)	184025 (0.70MB)	61341	469.17MB
SO ($\kappa = 1\%$)	122683392 (468MB)	2453667 (9.36MB)	3680501 (14.04MB)	1226833	491.40MB
SO ($\kappa = 2\%$)	122683392 (468MB)	4907335 (18.72MB)	7361003 (28.08MB)	2453667	514.80MB
SO ($\kappa = 5\%$)	122683392 (468MB)	12268339 (46.80MB)	18402508 (70.20MB)	6134169	585.00MB
SO ($\kappa = 8\%$)	122683392 (468MB)	19629342 (74.88MB)	29444014 (112.32MB)	9814671	655.20MB
SO ($\kappa = 10\%$)	122683392 (468MB)	24536678 (93.60MB)	36805017 (140.40MB)	12268339	702.00MB
GaLoRE ($r = 2$)	122683392 (468.00MB)	122683392 (468.00MB)	706560 (2.70MB)	122683392	938.70MB
GaLoRE ($r = 4$)	122683392 (468.00MB)	122683392 (468.00MB)	1413120 (5.39MB)	122683392	941.39MB
GaLoRE ($r = 8$)	122683392 (468.00MB)	122683392 (468.00MB)	2826240 (10.78MB)	122683392	946.78MB
GaLoRE ($r = 16$)	122683392 (468.00MB)	122683392 (468.00MB)	5652480 (21.56MB)	122683392	957.56MB
LoRA ($r = 2$)	123174912 (469.88MB)	491520 (1.88MB)	983040 (3.75MB)	491520	475.50MB
LoRA ($r = 4$)	123666432 (471.75MB)	983040 (3.75MB)	1966080 (7.50MB)	983040	483.00MB
LoRA ($r = 8$)	124649472 (475.50MB)	1966080 (7.50MB)	3932160 (15.00MB)	1966080	498.00MB
LoRA ($r = 16$)	126615552 (483.00MB)	3932160 (15.00MB)	7864320 (30.00MB)	3932160	528.00MB
PiSSA ($r = 2$)	123174912 (469.88MB)	491520 (1.88MB)	983040 (3.75MB)	491520	475.50MB
PiSSA ($r = 4$)	123666432 (471.75MB)	983040 (3.75MB)	1966080 (7.50MB)	983040	483.00MB
PiSSA ($r = 8$)	124649472 (475.50MB)	1966080 (7.50MB)	3932160 (15.00MB)	1966080	498.00MB
PiSSA ($r = 16$)	126615552 (483.00MB)	3932160 (15.00MB)	7864320 (30.00MB)	3932160	528.00MB
DoRA ($r = 2$)	123313152 (470.40MB)	629760 (2.40MB)	1259520 (4.80MB)	629760	477.61MB
DoRA ($r = 4$)	123804672 (472.28MB)	1121280 (4.28MB)	2242560 (8.55MB)	1121280	485.11MB
DoRA ($r = 8$)	124787712 (476.03MB)	2104320 (8.03MB)	4208640 (16.05MB)	2104320	500.11MB
DoRA ($r = 16$)	126753792 (483.53MB)	4070400 (15.53MB)	8140800 (31.05MB)	4070400	530.11MB
ReLoRA ($r = 2$)	123174912 (469.88MB)	491520 (1.88MB)	983040 (3.75MB)	491520	475.50MB
ReLoRA ($r = 4$)	123666432 (471.75MB)	983040 (3.75MB)	1966080 (7.50MB)	983040	483.00MB
ReLoRA ($r = 8$)	124649472 (475.50MB)	1966080 (7.50MB)	3932160 (15.00MB)	1966080	498.00MB
ReLoRA ($r = 16$)	126615552 (483.00MB)	3932160 (15.00MB)	7864320 (30.00MB)	3932160	528.00MB
VeRA ($r = 2$)	123313344 (470.40MB)	138432 (0.53MB)	276864 (1.06MB)	138432	471.99MB
VeRA ($r = 4$)	123805056 (472.28MB)	138624 (0.53MB)	277248 (1.06MB)	138624	473.87MB
VeRA ($r = 8$)	124788480 (476.03MB)	139008 (0.53MB)	278016 (1.06MB)	139008	477.62MB
VeRA ($r = 16$)	126755328 (483.53MB)	139776 (0.53MB)	279552 (1.07MB)	139776	485.13MB
Adam (Full Finetune)	122683392 (468.00MB)	122683392 (468.00MB)	245366784 (936.00MB)	122683392	1872.00MB

Table 3. Runtime on the ImageNet dataset using ViT-B/16.

	2 shots			4 shots		
Method	Iter.	Total	s/iter	Iter.	Total	s/iter
ADAM	2000	56 min.	1.69	4000	1h53	1.69
LoRA (r=2)	2000	52 min.	1.56	4000	1h43	1.55
SO (κ=0.05%)	2000	59 min.	1.76	4000	1h58	1.77

Table 4. Top-1 accuracy (%) on 4 datasets with ViT-L/14.

Dataset	1 shot		2 shots		4 shots	
	ReLoRA	SO	ReLoRA	SO	ReLoRA	SO
DTD	61.5	61.9	67.3	68.4	69.3	71.0
EuroSat	75.7	79.7	85.1	85.7	86.7	86.9
Aircraft	40.0	42.1	41.8	46.0	48.2	51.3
UCF	82.5	83.0	84.5	85.3	86.5	87.8

of $1 - \kappa$ with $\kappa = 0.05\%$. This extremely sparse update leads to minimal overheads in the gradient and optimizer states (≈ 0.47 MB and 0.70 MB, respectively). As a result, the total memory grows only slightly beyond the baseline weight storage. Rank-based techniques, by contrast, rely on separate low-rank matrices and typically require more memory than SO at extreme sparsities. Adam imposes the highest overhead due to storing a full gradient and two full optimizer states for every parameter.

Hence, even at very low κ (i.e., 0.05%), SO preserves adaption flexibility while significantly reducing memory consumption, which is particularly advantageous in few-shot or resource-constrained settings.

Training efficiency and scalability. SO updates only $\lfloor \kappa d \rfloor$ parameters per step and highly compresses the gradient and the moments ($\kappa = 0.05\%$). Table 3 of this document shows

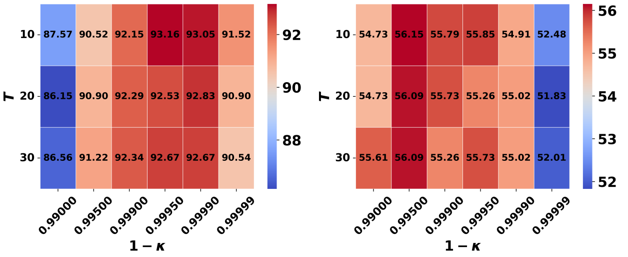
Algorithm 1 SO: Sparse Optimization Algorithm

```
Require: $\eta$ (learning rate), $\beta_1, \beta_2 \in [0, 1]$ (exponential decay rates for moment estimates), $\kappa$ (density ratio), $T$ (number of iterations before updating sparsity support), $\epsilon$ (numerical stability constant), $\tau$ (convergence rate)
Require: $\Theta_0$
$\tilde{\mu}_0 \leftarrow 0$
$\tilde{\nu}_0 \leftarrow 0$
$t \leftarrow 0$
$\mathfrak{I} \leftarrow \emptyset$ ($\mathfrak{I}$ denotes gradient sparsity support)
while $|\mathcal{L}(\Theta_{t-1})| > \tau$ do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\Theta \mathcal{L}(\Theta_{t-1})$
    $M \leftarrow \lfloor \kappa d \rfloor$
    if $(t-1) \mod T == 0$ then
        $\tilde{g}_t \leftarrow$ Random-$M(g_t)$
        $\mathfrak{I} \leftarrow \mathcal{I}(\tilde{g}_t)$
    else
        $\tilde{g}_t \leftarrow g_t[\mathfrak{I}]$
    end if
    $\mu_t \leftarrow \beta_1 \tilde{\mu}_{t-1} + (1 - \beta_1)\tilde{g}_t$
    $\nu_t \leftarrow \beta_2 \tilde{\nu}_{t-1} + (1 - \beta_2)\tilde{g}_t^2$
    if $(t-1) \mod T == 0$ then
        $\tilde{\mu}_t \leftarrow$ Top-$M(\mu_t)$
        $\tilde{\nu}_t \leftarrow \nu_t[\mathcal{I}(\tilde{\mu}_t)]$
    else
        $\tilde{\mu}_t \leftarrow \mu_t[\mathfrak{I}]$
        $\tilde{\nu}_t \leftarrow \nu_t[\mathfrak{I}]$
    end if
    $\hat{\mu}_t \leftarrow \frac{\tilde{\mu}_t}{1-\beta_1^t}$
    $\hat{\nu}_t \leftarrow \frac{\tilde{\nu}_t}{1-\beta_2^t}$
    $\Theta_t \leftarrow \Theta_{t-1} - \frac{\eta}{\sqrt{\hat{\nu}_t}+\epsilon}\hat{\mu}_t$
end while
return $\Theta_t$
```

the runtime of SO, LoRA, and Adam (2-shot and 4-shot finetuning on Imagenet). The runtime of SO is comparable to that of Adam and LoRA. To further test scalability, we finetuned ViT-L/14 (backbone size is 307M), which has approximately four times more parameters than ViT-B/16. As shown in Table 4, SO improved the performance compared with ReLoRA (second-best method in our comparison) by +1.8 (1-shot), +1.7 (2-shot), and +1.6 (4-shot) average top-1 accuracy over 4 datasets.

D. Sensitivity

Fig. 1 illustrates the impact of the density ratio κ and the update interval T on one-shot adaptation. In these experiments, we vary κ over a set of values $\{0.99,\ 0.995,\ 0.999,\ 0.9995,\ 0.9999,\ 0.9999\}$ and choose update intervals $T \in \{10,\ 20,\ 30\}$.



(a) Pets

(b) DTD

Figure 1. Test accuracy of SO for different density ratios κ and update intervals T (1-shot setting).

A smaller κ generally leads to improved performance as it prevents overfitting by retaining fewer parameter updates per step. However, overly low κ reduces the model's effective capacity and degrades accuracy. Similarly, smaller T values lead to better performance by promoting a more dynamic selection of trainable parameters.

E. Hardware and Software

All experiments are conducted on a Linux server with a consistent hardware and software environment. Table 5 provides details on the hardware and software used.

Table 5. Hardware and software.

Hardware	
RAM	504 GB
CPU model	Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz
# of CPUs	48
GPU model	NVIDIA RTX A6000
GPU memory	48 GB
# of GPUs	4
Software	
Operating System	Ubuntu 18.04.6 LTS
Python	3.10.16
PyTorch	2.5.1
CUDA	12.4

F. Additional Results

LoRA Pitfalls. Figures 2 and 3 illustrate LoRA's performance under few-shot settings, specifically with 2-shot and 4-shot. We finetune CLIP with a ViT-B/16 backbone on three benchmark datasets—DTD, Oxford Pets, and UCF101—and vary the LoRA rank in $\{2, 3, 4, 5\}$. We set the LoRA rank in $\{2, 3, 4, 5\}$ and train for a maximum of 2000 iterations or until the training loss reaches or falls below 0.01, whichever occurs first. At each iteration, we measure test-set accuracy to assess overfitting and convergence.

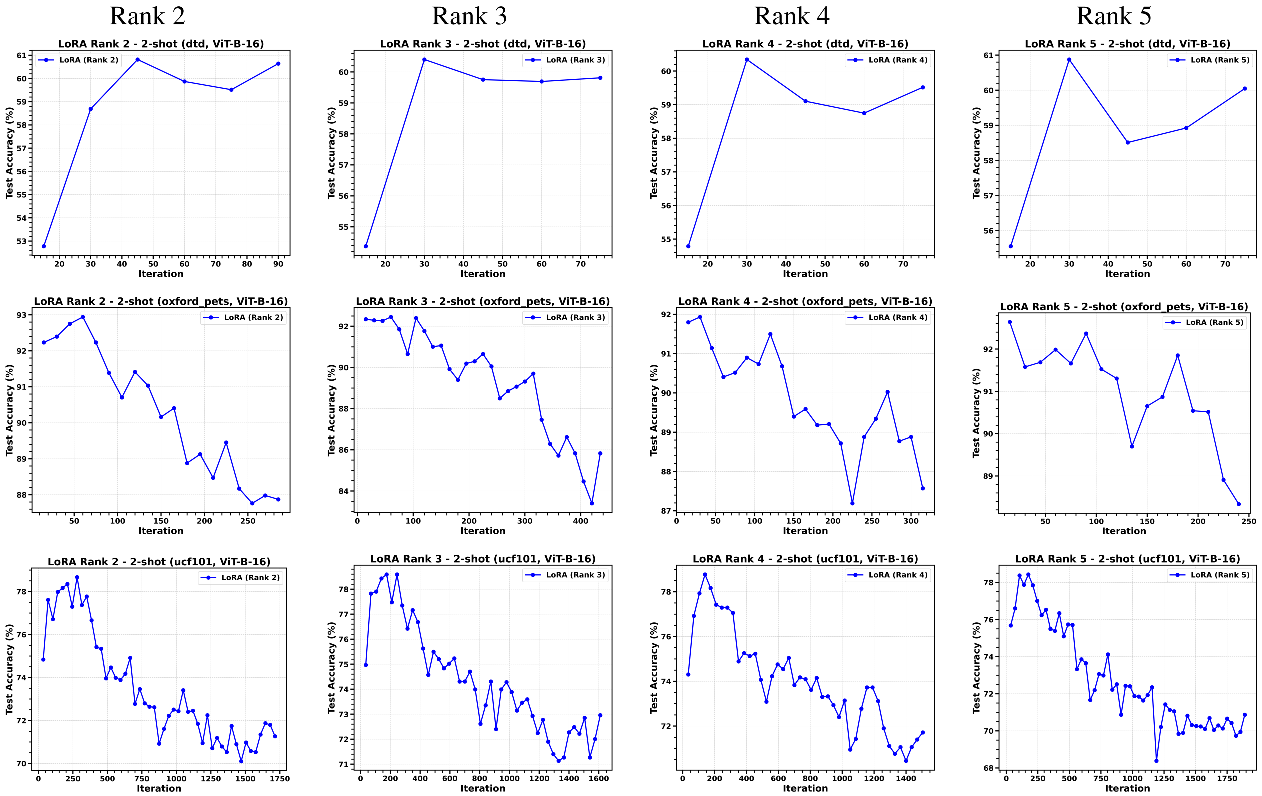


Figure 2. LoRA performance in a 2-shot setting on three datasets—DTD, Oxford Pets, and UCF101— using a pretrained CLIP with ViT-B/16 backbone. The model is trained for at most 2000 iterations or until the loss ≤ 0.01.

First, LoRA generally shows an improvement in test accuracy at the beginning of the training process, but results deteriorate noticeably as training proceeds. For instance, in the 2-shot Pets case at rank 2, accuracy peaks around 200 iterations before declining by over 5%, indicating overfitting. In contrast, the 4-shot DTD setting at rank 4 peaks around 300 iterations and then loses several points of accuracy once training proceeds.

Second, the optimal rank differs by dataset and shot setting. While rank 2 seems sufficient for DTD, it is not associated with the best results on the Pets or UCF101 datasets. The results of LoRA in 2-shot learning on UCF101 at rank 5 initially surpasses rank 2 but soon drop below it once the model overfits. These behaviors illustrate how peak accuracy depends strongly on the dataset, the number of shots, and the chosen rank.

Finally, the oscillatory accuracy trends underscore LoRA's sensitivity to both the rank parameter and the number of training iterations. Such fluctuations align with our main critique of LoRA. This method can be unstable in few-shot scenarios, which makes it difficult to choose a single hyperparameter setup that generalizes well across tasks.

Figures 4a and 4b illustrate SO's test accuracy across training iterations when using importance-based gradient pruning for 1-shot adaptation of a pretrained CLIP (ViT-B/16). We set the density ratio to $\kappa = 0.05\%$ and refresh the sparsity support every $T = 10$ iterations, training either until the loss falls below 0.01 or until 2000 iterations are reached.

On Oxford Pets (Fig. 4a), the model briefly attains nearly 93% accuracy before declining by about 7% due to overfitting. In UCF101 (Fig. 4b), accuracy rises above 74% but steadily drops and stabilizes near 64%. These trends confirm that, despite high initial gains, importance-based updates can still overfit in few-shot scenarios.

More Results. We include an 8/16-shot comparison with LoRA in Table 6. Furthermore, we implemented SAFT [1], a sparse optimization technique with a static support, following the paper. SO outperforms LoRA and SAFT across all few-shot settings. Unlike SO, SAFT uses a static sparsity support and importance-based gradient pruning, which may accelerate overfitting.

Applicability beyond CLIP Since SO is an optimizer built on Adam, it is model-agnostic. Besides the VLM experiments, we applied SO to BERT on RTE (GLUE) with 32-shot (see Table 7).

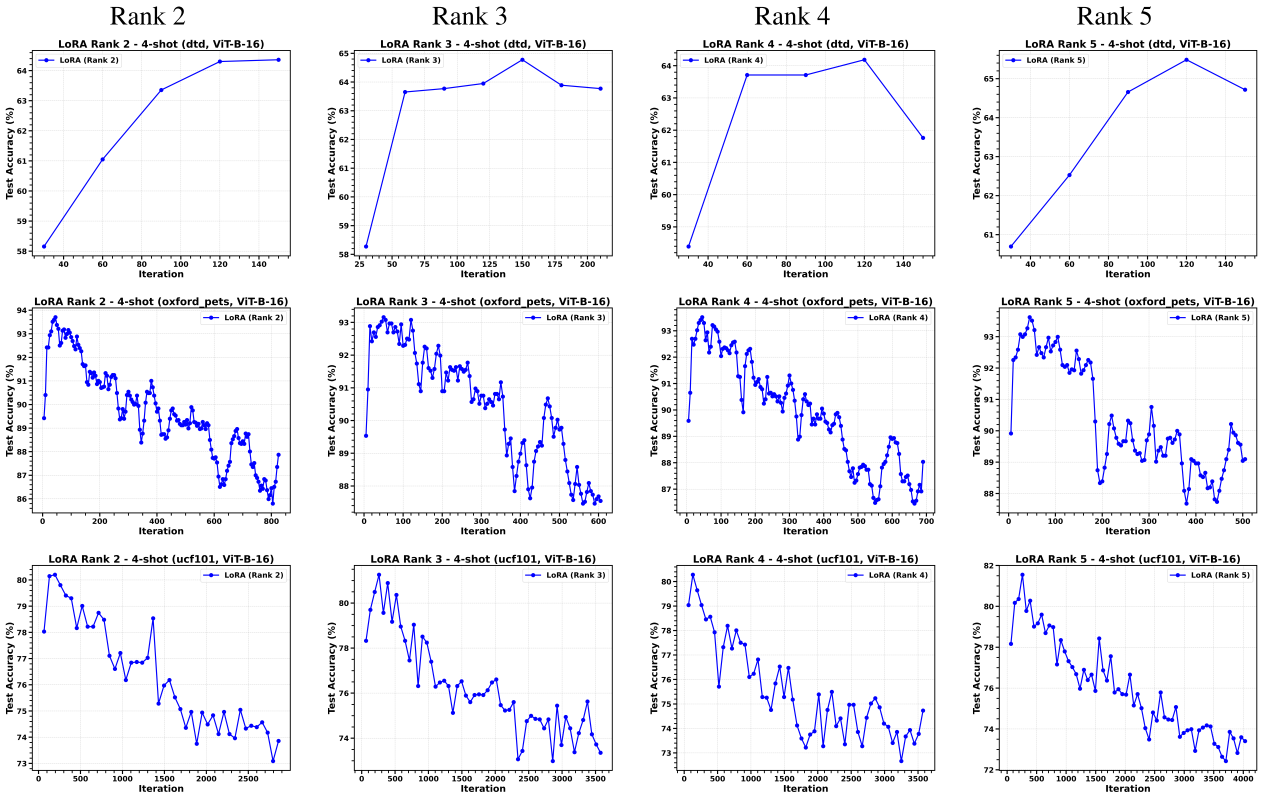


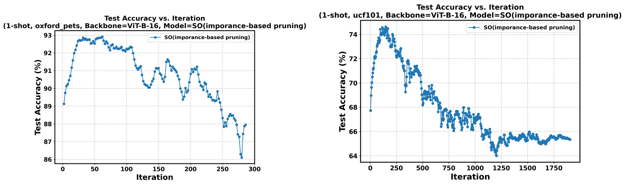
Figure 3. LoRA performance in a 4-shot setting on three datasets—DTD, Oxford Pets, and UCF101— using a pretrained CLIP with ViT-B/16 backbone. The model is trained for at most 2000 iterations or until the loss ≤ 0.01.

Table 6. Few-shot classification performance on 11 datasets with ViT-B/16 backbone. Top-1 accuracy (3 seeds); best in **bold**, second-best underlined.

Shots	Method	ImageNet	SUN	Aircraft	EuroSAT	Cars	Food	Pets	Flowers	Caltech	DTD	UCF	Average
0	CLIP (ICML '21)	66.7	62.6	24.7	47.5	65.3	86.1	89.1	71.4	92.9	43.6	66.7	65.1
1	LoRA (ICLR '22)	67.3	67.0	25.0	67.5	68.2	81.2	90.5	85.7	92.3	52.4	72.9	70.0
	SAFT (ECCV '24)	68.5	68.7	30.1	70.8	69.7	83.8	91.2	81.2	93.3	53.1	75.9	71.5
	SO (Ours)	70.1	70.3	31.5	78.2	71.6	86.2	93.3	84.9	94.1	55.3	76.4	73.8
2	LoRA (ICLR '22)	67.4	68.3	30.3	81.9	70.1	79.3	89.4	90.9	93.7	59.6	76.1	73.4
	SAFT (ECCV '24)	68.9	70.1	33.5	81.1	70.0	84.3	91.5	87.0	94.5	59.8	78.3	74.4
	SO (Ours)	70.5	72.3	37.2	82.7	74.4	85.3	92.2	91.9	95.3	60.2	80.4	76.6
4	LoRA (ICLR '22)	68.5	69.7	35.2	85.2	74.8	78.7	87.9	94.1	93.9	63.5	78.2	75.4
	SAFT (ECCV '24)	70.0	72.0	35.8	85.5	75.2	84.9	92.7	90.8	95.1	63.6	80.4	76.9
	SO (Ours)	71.4	73.7	38.6	87.7	78.9	85.3	92.4	95.1	95.5	66.4	83.4	78.9
8	LoRA (ICLR '22)	69.1	73.3	43.7	84.6	81.4	85.3	93.5	94.3	95.7	67.0	83.7	79.2
	SAFT (ECCV '24)	71.1	73.7	42.5	87.7	79.3	85.0	93.2	94.0	95.6	66.7	83.1	79.3
	SO (Ours)	72.2	72.7	45.6	87.5	82.8	85.4	93.6	95.6	95.8	67.6	83.9	**80.2**
16	LoRA (ICLR '22)	71.3	74.8	50.4	90.3	85.4	85.8	94.2	97.0	96.2	71.1	85.7	82.0
	SAFT (ECCV '24)	72.8	75.4	49.0	90.9	84.2	85.9	94.2	97.0	96.3	71.1	85.9	82.1
	SO (Ours)	73.3	74.0	54.7	92.6	86.8	85.7	94.3	97.7	96.4	72.8	86.5	**83.2**

G. Two-Layer Architecture

In addition to the experiments with CLIP, we evaluate SO on a simpler two-layer fully-connected network.



(a) Pets

(b) UCF

Figure 4. Test accuracy of SO (importance-based gradient pruning) with density ratio $\kappa = 0.05\%$ and update interval $T = 10$, applied to a pretrained CLIP (ViT-B/16) backbone. We train on a 1-shot setting for Pets and UCF101 until the loss < 0.01 or a maximum of 2000 iterations.

Table 7. Finetuning BERT (32-shot; stop when $\mathcal{L} \leq 10^{-3}$).

Task	LoRA				SO		
	$r = 2$	$r = 4$	$r = 8$	$r = 16$	$\kappa = 0.001\%$	$\kappa = 0.01\%$	$\kappa = 0.1\%$
RTE	54.5	54.2	54.2	54.5	57.0	57.0	55.6

G.1. Experimental Methodology

The two-layer fully connected network has an input layer of size 28×28, a single hidden layer of size 128 with ReLU activations, and an output layer matching the number of classes.

We conduct two types of experiments: **(i)** pretraining the model on MNIST or FMNIST, then fine-tuning on a target dataset (adaptation), and **(ii)** training from randomly initialized weights on the target dataset (no adaptation).

We explore both standard classification (full data) and few-shot learning (limited labeled data) under these scenarios. Finally, we include ablation studies that assess the influence of random-gradient pruning compared with importance-based gradient pruning.

All weights and biases are trainable, and we apply the same hyperparameters across all experiments for all methods. We train until convergence or for a maximum of 3000 iterations, whichever is reached first. In all two-layer network experiments, we use our SO optimizer and the low-rank baselines with the following default hyperparameters:

Hyperparameter	**Value**
Learning Rate	1×10^{-3}
Adam Betas	$(\beta_1, \beta_2) = (0.9,\ 0.999)$
Epsilon (ϵ)	1×10^{-8}
Sparsity Ratio (κ)	$[1\%, 2\%, 5\%, 8\%, 10\%]$
Update Interval (T)	30 (iterations)
Target Loss	10^{-4} (early stopping)

Table 8. Default hyperparameters for two-layer experiments.

We use the following six datasets for all experiments:

- **MNIST**: A canonical dataset of handwritten digits (0–9). Each sample is a 28×28 grayscale image, comprising 60k training and 10k test examples.
- **FashionMNIST (FMNIST)**: Contains 28×28 grayscale images of ten clothing and footwear categories. It serves as a harder drop-in replacement for MNIST in benchmarking.
- **EMNIST (Balanced Split)**: Extends MNIST to letters and digits, covering 47 classes of handwritten alphanumeric characters. It includes both uppercase and lowercase letters.
- **PathMNIST**: A medical image dataset with histopathological images of colorectal tissue, each labeled among nine classes (normal tissue, tumor tissue, etc.).
- **OrganMNISTAxial**: Features axial-view organ scans across eleven abdominal classes (e.g., spleen, kidney, aorta). Images are grayscale and resized to 28×28.
- **BloodMNIST**: Comprises microscopic blood cell images from eight categories. Samples are also converted to a standardized 28×28 resolution.

For adaptation, we first pretrain the models on MNIST or FMNIST. Then, we adapt this pretrained model to several target datasets. In each scenario, we compare SO to state-of-the-art low-rank methods (LoRA, ReLoRA, GaLoRE, etc.) and a fully finetuned baseline (Adam).

We measure classification accuracy on each target dataset. We report the mean and standard deviation over 10 runs with different random seeds. By presenting results for multiple tasks, we show the robustness and versatility of our sparse optimization approach compared to low-rank and dense baselines.

G.2. Memory Consumption

We evaluate GPU memory usage in a two-layer fully connected network. Tables 9 and 10 compare theoretical memory consumption, number of variables, and trainable parameters for the first and second layers, respectively. Activations and bias terms are excluded since they are identical for all methods.

Table 9 reports results for the first layer ($W_1 \in \mathbb{R}^{128 \times 784}$). Sparse Optimization (SO) requires fewer additional variables than most low-rank methods at a similar sparsity ratio. Its gradient and optimizer states are smaller because of dynamic sparsity. In contrast, Adam consumes the largest memory due to storing full gradients and optimizer states. Overall, SO balances memory and flexibility by freezing the base weights and updating a small subset of parameters.

Table 10 presents results for the second layer ($W_2 \in \mathbb{R}^{128 \times 128}$). SO uses considerably fewer gradients and optimizer variables, especially for low sparsity. Low-rank methods show higher memory usage due to extra low-rank matrices per layer. In contrast, SO's minimal updates mitigate memory overhead. Hence, the results confirm that SO

Table 9. Comparison of theoretical memory usage, number of variables, and trainable parameters for the first layer ($W_1 \in \mathbb{R}^{128\times784}$) of the two-layer fully connected architecture. Activations and bias terms are excluded because they remain consistent across all methods..

Method	Weight (#Vars, Mem.)	Gradient (#Vars, Mem.)	Opt. States (#Vars, Mem.)	#Trainable	Total Mem. (MB)
SO ($\kappa = 1\%$)	$100352\,(0.38MB)$	$2007\,(0.01MB)$	$3010\,(0.01MB)$	1003	0.40MB
SO ($\kappa = 2\%$)	$100352\,(0.38MB)$	$4014\,(0.02MB)$	$6021\,(0.02MB)$	2007	0.42MB
SO ($\kappa = 5\%$)	$100352\,(0.38MB)$	$10035\,(0.04MB)$	$15052\,(0.06MB)$	5017	0.48MB
SO ($\kappa = 8\%$)	$100352\,(0.38MB)$	$16056\,(0.06MB)$	$24084\,(0.09MB)$	8028	0.54MB
SO ($\kappa = 10\%$)	$100352\,(0.38MB)$	$20070\,(0.08MB)$	$30105\,(0.11MB)$	10035	0.57MB
GaLoRE ($r = 2$)	$100352\,(0.38MB)$	$100352\,(0.38MB)$	$3392\,(0.01MB)$	100352	0.78MB
GaLoRE ($r = 4$)	$100352\,(0.38MB)$	$100352\,(0.38MB)$	$6784\,(0.03MB)$	100352	0.79MB
GaLoRE ($r = 8$)	$100352\,(0.38MB)$	$100352\,(0.38MB)$	$13568\,(0.05MB)$	100352	0.82MB
GaLoRE ($r = 16$)	$100352\,(0.38MB)$	$100352\,(0.38MB)$	$27136\,(0.10MB)$	100352	0.87MB
LoRA ($r = 2$)	$102176\,(0.39MB)$	$1824\,(0.01MB)$	$3648\,(0.01MB)$	1824	0.41MB
LoRA ($r = 4$)	$104000\,(0.40MB)$	$3648\,(0.01MB)$	$7296\,(0.03MB)$	3648	0.44MB
LoRA ($r = 8$)	$107648\,(0.41MB)$	$7296\,(0.03MB)$	$14592\,(0.06MB)$	7296	0.49MB
LoRA ($r = 16$)	$114944\,(0.44MB)$	$14592\,(0.06MB)$	$29184\,(0.11MB)$	14592	0.61MB
PiSSA ($r = 2$)	$102176\,(0.39MB)$	$1824\,(0.01MB)$	$3648\,(0.01MB)$	1824	0.41MB
PiSSA ($r = 4$)	$104000\,(0.40MB)$	$3648\,(0.01MB)$	$7296\,(0.03MB)$	3648	0.44MB
PiSSA ($r = 8$)	$107648\,(0.41MB)$	$7296\,(0.03MB)$	$14592\,(0.06MB)$	7296	0.49MB
PiSSA ($r = 16$)	$114944\,(0.44MB)$	$14592\,(0.06MB)$	$29184\,(0.11MB)$	14592	0.61MB
DoRA ($r = 2$)	$102304\,(0.39MB)$	$1952\,(0.01MB)$	$3904\,(0.01MB)$	1952	0.41MB
DoRA ($r = 4$)	$104128\,(0.40MB)$	$3776\,(0.01MB)$	$7552\,(0.03MB)$	3776	0.44MB
DoRA ($r = 8$)	$107776\,(0.41MB)$	$7424\,(0.03MB)$	$14848\,(0.06MB)$	7424	0.50MB
DoRA ($r = 16$)	$115072\,(0.44MB)$	$14720\,(0.06MB)$	$29440\,(0.11MB)$	14720	0.61MB
ReLoRA ($r = 2$)	$102176\,(0.39MB)$	$1824\,(0.01MB)$	$3648\,(0.01MB)$	1824	0.41MB
ReLoRA ($r = 4$)	$104000\,(0.40MB)$	$3648\,(0.01MB)$	$7296\,(0.03MB)$	3648	0.44MB
ReLoRA ($r = 8$)	$107648\,(0.41MB)$	$7296\,(0.03MB)$	$14592\,(0.06MB)$	7296	0.49MB
ReLoRA ($r = 16$)	$114944\,(0.44MB)$	$14592\,(0.06MB)$	$29184\,(0.11MB)$	14592	0.61MB
VeRA ($r = 2$)	$102306\,(0.39MB)$	$130\,(0.00MB)$	$260\,(0.00MB)$	130	0.39MB
VeRA ($r = 4$)	$104132\,(0.40MB)$	$132\,(0.00MB)$	$264\,(0.00MB)$	132	0.40MB
VeRA ($r = 8$)	$107784\,(0.41MB)$	$136\,(0.00MB)$	$272\,(0.00MB)$	136	0.41MB
VeRA ($r = 16$)	$115088\,(0.44MB)$	$144\,(0.00MB)$	$288\,(0.00MB)$	144	0.44MB
Adam	$100352\,(0.38MB)$	$100352\,(0.38MB)$	$200704\,(0.77MB)$	100352	1.53MB

reduces memory requirements.

G.3. Results

Effectiveness in Classification. Table 11 presents classification performance when training from scratch (no pretraining) on each target dataset. Adam achieves slightly higher accuracy on some tasks (e.g., EMNIST, MNIST), but it often performs comparably or worse on others. Meanwhile, SO consistently outperforms GaLoRE ($r = 2$–16) on datasets like OrganMNISTAxial, BloodMNIST, and BreastMNIST. For instance, SO ($\kappa = 1\%$ or 2%) generally improves upon the low-rank methods while retaining fewer trainable parameters. This highlights the ability of optimizer.

Tables 12 and 13 report results after pretraining on MNIST or FMNIST, respectively. All methods benefit from pretraining, and Adam attains strong performance. However, our SO optimizer often yields higher or comparable accuracy to the low-rank baselines across most datasets, especially for moderate κ values (e.g., 1%–5%). In OrganMNISTAxial or BloodMNIST, for instance, SO frequently exceeds or matches GaLoRE and ReLoRA, while also preserving memory efficiency (Sec. G.2).

Tables 14, 15, and 16 compare two pruning strategies: importance-based (selecting the largest gradients) vs. randomness-based (selecting random gradients). When training from scratch or adapting a pretrained model, we observe that randomness-based pruning generally outperforms its importance-based counterpart, particularly at lower κ. This supports our hypothesis that sparse updates driven by random gradient selection mitigate overfitting more effectively than always choosing high-magnitude gradients.

Effectiveness in Few-Shot Learning. Table 17 shows results when training from scratch on just a few labeled samples per class (4, 8, and 16 shots). GaLoRE tends to underfit for low-shot regimes, especially when r is small. In contrast, SO ($\kappa \leq 2\%$) achieves higher accuracy on datasets such as EMNIST, MNIST, and BreastMNIST. For instance, at 4 shots, SO surpasses GaLoRE by up to 2–3% in EMNIST and BreastMNIST. This performance is probably attributed to the capacity of SO to mitigate overfitting in limited data situations, even without pretraining.

Tables 18 and 19 provide few-shot results after pretraining on MNIST or FMNIST, respectively. All methods improve considerably over the no-adaptation scenario, as the pretrained backbone offers a strong initialization. Nevertheless, SO still outperforms many low-rank baselines (LoRA, ReLoRA, GaLoRE) at 4, 8, or 16 shots, often matching or exceeding Adam. The benefits of sparsity persist in this setting, allowing SO to avoid overfitting.

Tables 20–22 compare importance-based vs. randomness-based gradient pruning in few-shot sce-

Table 10. Comparison of theoretical memory consumption, number of variables, and trainable parameters for the second layer ($W_2 \in \mathbb{R}^{128\times128}$) of the two-layer fully connected architecture. Activations and bias terms are excluded since they remain unchanged across methods.

Method	Weight (#Vars, Mem.)	Gradient (#Vars, Mem.)	Opt. States (#Vars, Mem.)	#Trainable	Total Mem. (MB)
SO ($\kappa = 0.01$)	16384 (0.06MB)	327 (0.00MB)	491 (0.00MB)	163	0.07MB
SO ($\kappa = 0.02$)	16384 (0.06MB)	655 (0.00MB)	983 (0.00MB)	327	0.07MB
SO ($\kappa = 0.05$)	16384 (0.06MB)	1638 (0.01MB)	2457 (0.01MB)	819	0.08MB
SO ($\kappa = 0.08$)	16384 (0.06MB)	2621 (0.01MB)	3932 (0.01MB)	1310	0.09MB
SO ($\kappa = 0.1$)	16384 (0.06MB)	3276 (0.01MB)	4915 (0.02MB)	1638	0.09MB
GaLoRE ($r = 2$)	16384 (0.06MB)	16384 (0.06MB)	768 (0.00MB)	16384	0.13MB
GaLoRE ($r = 4$)	16384 (0.06MB)	16384 (0.06MB)	1536 (0.01MB)	16384	0.13MB
GaLoRE ($r = 8$)	16384 (0.06MB)	16384 (0.06MB)	3072 (0.01MB)	16384	0.14MB
GaLoRE ($r = 16$)	16384 (0.06MB)	16384 (0.06MB)	6144 (0.02MB)	16384	0.15MB
LoRA ($r = 2$)	16896 (0.06MB)	512 (0.00MB)	1024 (0.00MB)	512	0.07MB
LoRA ($r = 4$)	17408 (0.07MB)	1024 (0.00MB)	2048 (0.01MB)	1024	0.08MB
LoRA ($r = 8$)	18432 (0.07MB)	2048 (0.01MB)	4096 (0.02MB)	2048	0.09MB
LoRA ($r = 16$)	20480 (0.08MB)	4096 (0.02MB)	8192 (0.03MB)	4096	0.12MB
PiSSA ($r = 2$)	16896 (0.06MB)	512 (0.00MB)	1024 (0.00MB)	512	0.07MB
PiSSA ($r = 4$)	17408 (0.07MB)	1024 (0.00MB)	2048 (0.01MB)	1024	0.08MB
PiSSA ($r = 8$)	18432 (0.07MB)	2048 (0.01MB)	4096 (0.02MB)	2048	0.09MB
PiSSA ($r = 16$)	20480 (0.08MB)	4096 (0.02MB)	8192 (0.03MB)	4096	0.12MB
DoRA ($r = 2$)	17024 (0.06MB)	640 (0.00MB)	1280 (0.00MB)	640	0.07MB
DoRA ($r = 4$)	17536 (0.07MB)	1152 (0.00MB)	2304 (0.01MB)	1152	0.08MB
DoRA ($r = 8$)	18560 (0.07MB)	2176 (0.01MB)	4352 (0.02MB)	2176	0.10MB
DoRA ($r = 16$)	20608 (0.08MB)	4224 (0.02MB)	8448 (0.03MB)	4224	0.13MB
ReLoRA ($r = 2$)	16896 (0.06MB)	512 (0.00MB)	1024 (0.00MB)	512	0.07MB
ReLoRA ($r = 4$)	17408 (0.07MB)	1024 (0.00MB)	2048 (0.01MB)	1024	0.08MB
ReLoRA ($r = 8$)	18432 (0.07MB)	2048 (0.01MB)	4096 (0.02MB)	2048	0.09MB
ReLoRA ($r = 16$)	20480 (0.08MB)	4096 (0.02MB)	8192 (0.03MB)	4096	0.12MB
VeRA ($r = 2$)	17026 (0.06MB)	130 (0.00MB)	260 (0.00MB)	130	0.07MB
VeRA ($r = 4$)	17540 (0.07MB)	132 (0.00MB)	264 (0.00MB)	132	0.07MB
VeRA ($r = 8$)	18568 (0.07MB)	136 (0.00MB)	272 (0.00MB)	136	0.07MB
VeRA ($r = 16$)	20624 (0.08MB)	144 (0.00MB)	288 (0.00MB)	144	0.08MB
Adam	16384 (0.06MB)	16384 (0.06MB)	32768 (0.12MB)	16384	0.25MB

narios. We observe that random gradient selection provides better accuracy, especially at lower κ. By avoiding exclusive reliance on large-magnitude updates, SO's sparse updates reduce overfitting risk.

Full-Rank Learning. Figures 5–16 illustrate how random gradient pruning maintains a high-dimensional update space, effectively enabling full-rank learning despite extreme sparsity.

In rank evolution plots, the gradient rank for random pruning remains close to the full rank throughout training, whereas the gradient rank for importance-based pruning often settles to a lower value. This outcome suggests that the random selection of gradient entries explores more diverse directions in parameter space, thereby preserving expressive capacity. Further, the loss curves confirm that random pruning converges stably and less rapidly, while importance-based pruning risks collapsing updates into fewer directions, potentially causing overfitting. Random gradient pruning causes a slow learning process but is more stable and less prone to overfitting, while importance-based gradient learning leads to fast learning and potential overfitting.

Overall, these figures illustrate that sparsity —even at very low-density ratios— does not diminish the fundamental rank of the gradient and previous results confirm that sparsity does not lower the learning capacity.

References

[1] Bac Nguyen, Stefan Uhlich, Fabien Cardinaux, Lukas Mauch, Marzieh Edraki, and Aaron Courville. Saft: Towards out-of-distribution generalization in fine-tuning. In *European Conference on Computer Vision (ECCV)*, pages 138–154, 2024. 4

Table 11. Classication performance on 7 datasets with a two-layer fully-connected architecture. Results are the average top-1 accuracy of 10 executions ± standard deviation.

Model	**EMNIST**	**MNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
GaLoRE ($r = 2$)	80.64 ±0.17	96.06 ±0.12	86.35 ±0.25	49.41 ±0.97	67.85 ±0.48	75.57 ±0.35	77.37 ±1.35
GaLoRE ($r = 4$)	80.80 ±0.16	95.89 ±0.12	86.31 ±0.20	53.27 ±0.91	68.32 ±0.44	76.01 ±0.22	77.88 ±0.87
GaLoRE ($r = 8$)	81.29 ±0.25	96.04 ±0.16	86.68 ±0.28	53.74 ±0.94	69.19 ±0.38	77.54 ±0.32	78.65 ±1.28
GaLoRE ($r = 16$)	81.74 ±0.28	97.07 ±0.13	87.48 ±0.27	53.73 ±1.44	70.56 ±0.40	77.96 ±0.38	78.53 ±1.47
SO ($\kappa = 1\%$)	81.29 ±0.18	97.16 ±0.12	87.46 ±0.19	53.63 ±0.64	72.67 ±0.35	78.89 ±0.42	79.62 ±1.06
SO ($\kappa = 2\%$)	81.55 ±0.16	97.18 ±0.10	87.67 ±0.28	53.43 ±0.82	72.19 ±0.61	79.14 ±0.45	78.78 ±1.16
SO ($\kappa = 5\%$)	81.63 ±0.25	97.24 ±0.09	87.66 ±0.36	54.41 ±0.89	72.11 ±0.41	78.81 ±0.37	78.65 ±1.79
SO ($\kappa = 8\%$)	81.50 ±0.31	97.32 ±0.14	87.52 ±0.30	54.20 ±1.02	71.85 ±0.45	78.74 ±0.60	77.76 ±1.00
SO ($\kappa = 10\%$)	81.40 ±0.38	97.36 ±0.18	87.78 ±0.26	52.65 ±1.06	71.74 ±0.56	78.38 ±0.41	77.31 ±1.50
Adam	82.19 ±0.53	97.53 ±0.14	88.08 ±0.21	52.19 ±1.25	72.38 ±0.47	78.49 ±0.38	77.56 ±0.86

Table 12. Classication performance on 6 target datasets after pretraining on MNIST with a two-layer fully-connected architecture. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Model	**EMNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
LoRA ($r = 2$)	75.85 ±0.29	84.96 ±0.18	51.24 ±0.73	67.11 ±0.81	76.06 ±0.18	77.37 ±1.32
LoRA ($r = 4$)	78.51 ±0.16	85.93 ±0.26	52.03 ±0.77	69.13 ±0.62	77.09 ±0.56	78.21 ±1.15
LoRA ($r = 8$)	80.52 ±0.15	86.68 ±0.28	52.28 ±0.94	71.70 ±0.53	77.76 ±0.24	77.95 ±0.92
LoRA ($r = 16$)	81.82 ±0.24	87.03 ±0.26	52.58 ±1.36	71.92 ±0.46	78.00 ±0.36	77.63 ±1.26
ReLoRA ($r = 2$)	78.67 ±0.17	85.58 ±0.20	50.19 ±0.60	70.62 ±0.37	75.79 ±0.44	78.21 ±1.25
ReLoRA ($r = 4$)	78.76 ±0.18	85.65 ±0.20	50.43 ±0.43	70.65 ±0.45	76.90 ±0.47	78.01 ±1.32
ReLoRA ($r = 8$)	78.73 ±0.17	85.60 ±0.11	50.62 ±0.45	70.80 ±0.55	78.03 ±0.45	77.82 ±1.52
ReLoRA ($r = 16$)	78.81 ±0.16	85.56 ±0.14	50.54 ±0.56	70.55 ±0.49	78.20 ±0.41	77.44 ±1.37
GaLoRE ($r = 2$)	80.26 ±0.23	86.40 ±0.23	50.50 ±0.70	70.71 ±0.38	76.43 ±0.60	78.53 ±0.72
GaLoRE ($r = 4$)	81.02 ±0.30	86.71 ±0.21	50.79 ±0.93	71.44 ±0.41	77.24 ±0.39	80.45 ±1.86
GaLoRE ($r = 8$)	81.60 ±0.19	87.08 ±0.21	51.36 ±0.77	71.98 ±0.43	77.97 ±0.26	82.56 ±0.94
GaLoRE ($r = 16$)	82.04 ±0.22	87.63 ±0.23	52.02 ±0.79	71.95 ±0.54	78.84 ±0.45	82.56 ±1.31
SO ($\kappa = 1\%$)	81.64 ±0.25	87.55 ±0.21	52.17 ±0.94	74.34 ±0.69	79.21 ±0.25	82.31 ±1.32
SO ($\kappa = 2\%$)	81.82 ±0.09	87.64 ±0.14	52.91 ±0.90	74.20 ±0.22	79.56 ±0.31	82.18 ±1.43
SO ($\kappa = 5\%$)	81.99 ±0.11	87.68 ±0.23	53.14 ±1.00	73.67 ±0.47	79.76 ±0.53	82.37 ±1.38
SO ($\kappa = 8\%$)	81.82 ±0.27	87.74 ±0.21	53.18 ±1.23	73.64 ±0.56	79.66 ±0.43	81.99 ±1.23
SO ($\kappa = 10\%$)	81.73 ±0.23	87.55 ±0.31	52.97 ±1.11	73.20 ±0.47	79.47 ±0.60	80.77 ±1.03
Adam	82.47 ±0.35	88.03 ±0.17	51.59 ±0.92	73.23 ±0.42	79.42 ±0.51	82.05 ±1.34

Table 13. Classification performance on 6 target datasets after pretraining on FMNIST with a two-layer fully-connected architecture. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Model	**EMNIST**	**MNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
LoRA ($r = 2$)	74.83 ±0.18	94.80 ±0.25	50.69 ±0.70	68.14 ±0.34	74.89 ±0.46	77.95 ±1.93
LoRA ($r = 4$)	77.44 ±0.20	95.73 ±0.15	50.43 ±0.54	69.97 ±0.41	76.40 ±0.21	78.21 ±1.15
LoRA ($r = 8$)	79.59 ±0.16	96.32 ±0.12	50.73 ±1.09	70.73 ±0.26	76.30 ±0.58	78.59 ±0.96
LoRA ($r = 16$)	81.18 ±0.14	96.48 ±0.16	51.52 ±0.61	71.17 ±0.55	76.89 ±0.66	78.65 ±1.46
ReLoRA ($r = 2$)	77.79 ±0.18	95.39 ±0.12	49.99 ±0.39	70.63 ±0.38	74.93 ±0.44	78.01 ±1.55
ReLoRA ($r = 4$)	77.74 ±0.24	95.29 ±0.14	50.01 ±0.59	70.49 ±0.37	75.97 ±0.48	78.14 ±0.97
ReLoRA ($r = 8$)	77.76 ±0.22	95.36 ±0.10	50.18 ±0.62	70.62 ±0.42	76.45 ±0.48	79.42 ±1.61
ReLoRA ($r = 16$)	77.79 ±0.20	95.44 ±0.13	49.82 ±0.40	70.70 ±0.35	76.59 ±0.51	78.27 ±1.53
GaLoRE ($r = 2$)	80.00 ±0.23	95.86 ±0.16	50.21 ±0.56	70.64 ±0.36	75.64 ±0.46	79.68 ±1.46
GaLoRE ($r = 4$)	80.41 ±0.36	96.15 ±0.16	50.46 ±0.90	71.15 ±0.59	76.55 ±0.46	78.91 ±1.47
GaLoRE ($r = 8$)	81.07 ±0.24	96.53 ±0.08	51.84 ±0.62	71.23 ±0.43	77.66 ±0.30	79.55 ±1.73
GaLoRE ($r = 16$)	81.61 ±0.21	97.08 ±0.20	52.20 ±0.61	71.99 ±0.57	78.48 ±0.40	79.62 ±2.62
SO ($\kappa = 1\%$)	81.17 ±0.16	97.07 ±0.13	52.14 ±0.59	72.94 ±0.35	77.64 ±0.41	79.94 ±1.41
SO ($\kappa = 2\%$)	81.33 ±0.25	97.32 ±0.08	52.82 ±0.49	73.39 ±0.28	78.13 ±0.35	80.13 ±1.07
SO ($\kappa = 5\%$)	81.58 ±0.29	97.47 ±0.13	53.22 ±0.77	73.23 ±0.55	78.55 ±0.43	80.58 ±1.92
SO ($\kappa = 8\%$)	81.58 ±0.34	97.44 ±0.09	53.41 ±1.05	72.93 ±0.37	78.47 ±0.53	80.00 ±1.40
SO ($\kappa = 10\%$)	81.35 ±0.32	97.43 ±0.10	53.27 ±1.17	73.07 ±0.37	78.42 ±0.47	79.94 ±1.11
Adam	82.08 ±0.30	97.69 ±0.11	52.36 ±0.69	72.88 ±0.56	79.04 ±0.43	80.71 ±1.26

Table 14. Classification performance on 7 datasets with a two-layer fully-connected architecture. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Strategy	**EMNIST**	**MNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
Importance-Based Gradient Pruning							
SO ($\kappa = 1\%$)	76.81 ±0.45	95.05 ±0.44	85.42 ±0.38	50.22 ±1.23	67.34 ±0.84	74.60 ±0.72	77.63 ±2.28
SO ($\kappa = 2\%$)	78.52 ±0.78	95.46 ±0.37	86.29 ±0.37	52.18 ±0.48	67.90 ±0.31	75.18 ±0.59	79.55 ±1.56
SO ($\kappa = 5\%$)	80.05 ±0.23	96.27 ±0.26	86.70 ±0.20	53.19 ±0.77	68.17 ±0.75	75.68 ±0.62	78.53 ±1.44
SO ($\kappa = 8\%$)	80.59 ±0.38	96.51 ±0.20	86.94 ±0.33	53.39 ±0.76	69.33 ±0.66	75.88 ±0.92	76.92 ±2.48
SO ($\kappa = 10\%$)	80.20 ±0.45	96.55 ±0.20	87.30 ±0.34	53.65 ±0.50	69.89 ±0.55	76.35 ±0.34	77.05 ±6.52
Random Gradient Pruning							
SO ($\kappa = 1\%$)	81.29 ±0.18	97.16 ±0.12	87.46 ±0.19	53.63 ±0.64	72.67 ±0.35	78.89 ±0.42	79.62 ±1.06
SO ($\kappa = 2\%$)	81.55 ±0.16	97.18 ±0.10	87.67 ±0.28	53.43 ±0.82	72.19 ±0.61	79.14 ±0.45	78.78 ±1.16
SO ($\kappa = 5\%$)	81.63 ±0.25	97.24 ±0.09	87.66 ±0.36	54.41 ±0.89	72.11 ±0.41	78.81 ±0.37	78.65 ±1.79
SO ($\kappa = 8\%$)	81.50 ±0.31	97.32 ±0.14	87.52 ±0.30	54.20 ±1.02	71.85 ±0.45	78.74 ±0.60	77.76 ±1.00
SO ($\kappa = 10\%$)	81.40 ±0.38	97.36 ±0.18	87.78 ±0.26	52.65 ±1.06	71.74 ±0.56	78.38 ±0.41	77.31 ±1.50

Table 15. Classification performance on 6 target datasets after pretraining on MNIST with a two-layer fully-connected architecture. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Strategy	**EMNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
Importance-Based Gradient Pruning						
SO ($\kappa = 1\%$)	79.89 ±0.26	86.54 ±0.34	51.42 ±1.51	69.82 ±0.46	76.09 ±0.42	81.35 ±1.89
SO ($\kappa = 2\%$)	80.12 ±0.33	86.54 ±0.38	51.15 ±2.15	69.95 ±0.80	76.17 ±0.47	81.35 ±1.56
SO ($\kappa = 5\%$)	80.23 ±0.45	87.08 ±0.28	51.54 ±1.75	70.60 ±0.68	76.45 ±0.55	81.41 ±1.22
SO ($\kappa = 8\%$)	80.50 ±0.41	87.41 ±0.28	53.23 ±1.22	71.01 ±0.53	77.34 ±0.50	82.05 ±1.81
SO ($\kappa = 10\%$)	80.80 ±0.35	87.14 ±0.30	53.66 ±0.87	71.08 ±0.57	77.27 ±0.77	80.77 ±2.20
Random Gradient Pruning						
SO ($\kappa = 1\%$)	81.64 ±0.25	87.55 ±0.21	52.17 ±0.94	74.34 ±0.69	79.21 ±0.25	82.31 ±1.32
SO ($\kappa = 2\%$)	81.82 ±0.09	87.64 ±0.14	52.91 ±0.90	74.20 ±0.22	79.56 ±0.31	82.18 ±1.43
SO ($\kappa = 5\%$)	81.99 ±0.11	87.68 ±0.23	53.14 ±1.00	73.67 ±0.47	79.76 ±0.53	82.37 ±1.38
SO ($\kappa = 8\%$)	81.82 ±0.27	87.74 ±0.21	53.18 ±1.23	73.64 ±0.56	79.66 ±0.43	81.99 ±1.23
SO ($\kappa = 10\%$)	81.73 ±0.23	87.55 ±0.31	52.97 ±1.11	73.20 ±0.47	79.47 ±0.60	80.77 ±1.03

Table 16. Classication performance on 6 target datasets after pretraining on FMNIST with a two-layer fully-connected architecture. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Strategy	**EMNIST**	**MNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
Importance-Based Gradient Pruning						
SO ($\kappa = 1\%$)	79.13 ±0.45	96.05 ±0.16	51.80 ±0.92	70.15 ±0.53	75.95 ±0.60	79.42 ±2.04
SO ($\kappa = 2\%$)	79.58 ±0.36	96.16 ±0.14	52.25 ±0.95	70.23 ±0.55	76.00 ±0.89	79.04 ±1.81
SO ($\kappa = 5\%$)	80.21 ±0.32	96.27 ±0.41	52.13 ±2.10	70.33 ±0.54	76.81 ±0.61	79.62 ±1.90
SO ($\kappa = 8\%$)	80.42 ±0.42	96.60 ±0.20	52.26 ±1.24	71.05 ±0.77	76.83 ±0.96	78.59 ±1.72
SO ($\kappa = 10\%$)	80.69 ±0.30	96.71 ±0.40	53.64 ±0.93	71.50 ±0.60	77.20 ±0.66	79.29 ±1.77
Random Gradient Pruning						
SO ($\kappa = 1\%$)	81.17 ±0.16	97.07 ±0.13	52.14 ±0.59	72.94 ±0.35	77.64 ±0.41	79.94 ±1.41
SO ($\kappa = 2\%$)	81.33 ±0.25	97.32 ±0.08	52.82 ±0.49	73.39 ±0.28	78.13 ±0.35	80.13 ±1.07
SO ($\kappa = 5\%$)	81.58 ±0.29	97.47 ±0.13	53.22 ±0.77	73.23 ±0.55	78.55 ±0.43	80.58 ±1.92
SO ($\kappa = 8\%$)	81.58 ±0.34	97.44 ±0.09	53.41 ±1.05	72.93 ±0.37	78.47 ±0.53	80.00 ±1.40
SO ($\kappa = 10\%$)	81.35 ±0.32	97.43 ±0.10	53.27 ±1.17	73.07 ±0.37	78.42 ±0.47	79.94 ±1.11

Table 17. Few-shot classification performance on 7 datasets using a two-layer fully-connected architecture without pretraining. Results are the average top-1 accuracy of 10 executions ± standard deviation.

Shots	**Model**	**EMNIST**	**MNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
4	GaLoRE ($r = 2$)	28.71 ±1.75	54.79 ±3.09	59.38 ±2.33	26.20 ±3.54	39.49 ±1.59	47.75 ±3.64	58.46 ±9.95
	GaLoRE ($r = 4$)	27.12 ±1.85	56.71 ±3.98	58.99 ±2.96	23.66 ±3.23	37.90 ±2.45	46.08 ±2.46	57.82 ±10.08
	GaLoRE ($r = 8$)	28.94 ±1.82	62.78 ±3.06	60.57 ±2.48	22.26 ±2.99	39.59 ±1.55	43.85 ±3.26	58.33 ±10.47
	GaLoRE ($r = 16$)	32.77 ±1.51	62.80 ±3.03	61.56 ±2.48	21.49 ±3.13	40.37 ±2.03	44.76 ±2.98	57.76 ±10.32
	SO ($\kappa = 1\%$)	34.17 ±1.83	63.20 ±3.03	61.15 ±2.78	26.61 ±5.09	40.72 ±1.79	46.47 ±2.64	57.88 ±11.01
	SO ($\kappa = 2\%$)	34.38 ±1.81	63.14 ±3.10	61.32 ±2.79	25.77 ±4.08	41.25 ±2.33	45.94 ±2.47	57.12 ±10.52
	SO ($\kappa = 5\%$)	34.07 ±1.84	62.98 ±2.52	61.81 ±2.21	22.93 ±4.88	40.92 ±2.37	46.94 ±2.50	56.86 ±10.26
	SO ($\kappa = 8\%$)	33.93 ±1.73	63.59 ±3.10	61.60 ±2.40	24.30 ±4.84	40.87 ±1.91	46.74 ±2.51	57.12 ±9.87
	SO ($\kappa = 10\%$)	34.22 ±1.80	63.02 ±3.32	61.16 ±2.22	21.27 ±2.96	40.08 ±1.87	45.42 ±2.24	58.46 ±10.54
	Adam	33.71 ±1.98	63.73 ±2.64	61.07 ±2.07	20.77 ±3.14	39.42 ±2.04	47.64 ±2.54	58.0 ±10.40
8	GaLoRE ($r = 2$)	35.20 ±1.28	66.16 ±3.45	64.66 ±1.66	25.52 ±3.48	43.77 ±1.50	51.21 ±3.74	63.01 ±9.18
	GaLoRE ($r = 4$)	33.85 ±1.83	66.06 ±3.11	65.51 ±1.86	23.38 ±2.68	43.19 ±2.24	51.04 ±3.86	63.14 ±8.63
	GaLoRE ($r = 8$)	35.71 ±1.25	71.64 ±2.48	67.18 ±1.59	22.47 ±3.38	44.18 ±2.03	49.01 ±2.66	62.82 ±8.43
	GaLoRE ($r = 16$)	40.98 ±1.15	73.00 ±2.00	67.50 ±1.10	21.96 ±3.50	45.31 ±2.05	49.39 ±3.10	63.78 ±8.73
	SO ($\kappa = 1\%$)	41.79 ±1.45	72.09 ±2.36	67.18 ±1.22	26.79 ±4.63	46.65 ±1.47	51.66 ±3.33	64.10 ±8.19
	SO ($\kappa = 2\%$)	41.78 ±1.27	72.34 ±2.48	67.42 ±1.32	25.39 ±3.78	46.66 ±2.43	51.66 ±2.96	63.46 ±7.57
	SO ($\kappa = 5\%$)	41.81 ±1.37	72.22 ±2.72	67.20 ±1.38	24.08 ±3.75	46.36 ±1.72	52.03 ±3.40	63.33 ±8.64
	SO ($\kappa = 8\%$)	41.69 ±1.19	71.86 ±2.47	67.11 ±1.24	23.33 ±3.63	45.87 ±1.87	52.16 ±3.60	63.59 ±8.34
	SO ($\kappa = 10\%$)	41.31 ±1.58	72.52 ±2.37	66.89 ±1.74	23.14 ±2.56	45.98 ±1.99	51.98 ±3.25	63.27 ±7.51
	Adam	41.93 ±1.34	72.88 ±2.38	67.37 ±1.54	20.79 ±2.32	44.85 ±1.91	52.81 ±3.21	63.01 ±10.13
16	GaLoRE ($r = 2$)	43.66 ±0.74	73.27 ±2.72	68.89 ±1.38	24.87 ±3.11	48.89 ±1.52	54.36 ±2.36	62.31 ±7.12
	GaLoRE ($r = 4$)	41.50 ±0.72	74.20 ±2.57	69.65 ±1.58	23.84 ±2.17	48.26 ±1.36	54.44 ±1.97	61.41 ±7.74
	GaLoRE ($r = 8$)	42.64 ±0.85	77.87 ±1.71	71.47 ±1.34	23.48 ±1.62	49.17 ±1.84	52.40 ±1.66	61.09 ±8.85
	GaLoRE ($r = 16$)	47.66 ±0.68	79.31 ±1.53	72.26 ±1.25	22.90 ±2.21	51.03 ±1.89	54.16 ±1.34	60.90 ±8.58
	SO ($\kappa = 1\%$)	48.60 ±0.68	78.99 ±1.39	71.94 ±1.57	27.21 ±2.55	52.85 ±1.19	55.93 ±1.68	62.12 ±9.50
	SO ($\kappa = 2\%$)	48.38 ±0.49	78.64 ±1.66	71.90 ±1.24	26.77 ±2.35	52.15 ±1.43	55.51 ±2.19	62.69 ±8.24
	SO ($\kappa = 5\%$)	48.45 ±0.82	79.45 ±1.46	71.44 ±1.66	27.06 ±2.65	51.77 ±1.62	56.38 ±1.80	61.15 ±9.45
	SO ($\kappa = 8\%$)	48.74 ±0.71	79.17 ±1.58	71.79 ±1.43	25.25 ±1.84	51.71 ±1.12	56.71 ±1.81	63.97 ±8.25
	SO ($\kappa = 10\%$)	48.77 ±1.07	79.06 ±1.53	71.90 ±1.34	25.50 ±2.39	51.77 ±1.19	56.81 ±1.58	60.51 ±10.17
	Adam	48.80 ±0.60	79.29 ±1.59	71.97 ±1.48	22.98 ±1.18	50.76 ±1.48	56.24 ±2.39	61.54 ±6.43

Table 18. Few-shot classification performance on 6 datasets using a two-layer fully-connected architecture after pretraining on MNIST. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Shots	**Model**	**EMNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
4	LoRA ($r = 2$)	28.02 ±1.32	55.36 ±2.30	26.69 ±2.21	31.73 ±2.66	45.21 ±4.09	61.28 ±7.25
	LoRA ($r = 4$)	28.44 ±1.32	55.21 ±2.97	26.81 ±2.45	34.10 ±2.05	45.79 ±2.60	60.71 ±8.61
	LoRA ($r = 8$)	29.08 ±1.23	56.70 ±2.62	27.50 ±2.72	34.87 ±1.70	45.41 ±2.47	60.13 ±8.21
	LoRA ($r = 16$)	29.14 ±1.42	55.60 ±2.80	27.19 ±1.61	34.74 ±1.12	46.23 ±3.38	60.19 ±8.96
	ReLoRA ($r = 2$)	27.84 ±1.42	55.36 ±2.30	26.69 ±2.21	31.73 ±2.66	44.61 ±3.36	60.90 ±9.48
	ReLoRA ($r = 4$)	28.28 ±1.45	55.21 ±2.97	26.81 ±2.45	34.10 ±2.05	45.52 ±2.29	60.64 ±7.82
	ReLoRA ($r = 8$)	28.89 ±1.32	56.70 ±2.62	27.50 ±2.72	34.87 ±1.70	45.14 ±3.23	59.87 ±7.60
	ReLoRA ($r = 16$)	29.07 ±1.43	55.60 ±2.80	27.19 ±1.61	34.74 ±1.12	45.71 ±2.90	59.29 ±8.32
	GaLoRE ($r = 2$)	29.86 ±1.38	56.93 ±2.74	26.23 ±2.91	33.46 ±1.89	44.25 ±3.43	59.81 ±8.43
	GaLoRE ($r = 4$)	30.47 ±1.48	58.63 ±2.46	25.13 ±2.36	34.30 ±1.00	45.88 ±2.69	60.00 ±7.70
	GaLoRE ($r = 8$)	30.89 ±1.65	59.58 ±2.63	23.19 ±1.68	35.46 ±2.05	44.42 ±3.30	61.99 ±8.81
	GaLoRE ($r = 16$)	31.87 ±1.61	58.98 ±2.78	23.09 ±2.16	35.19 ±0.91	43.82 ±2.78	60.90 ±9.55
	SO ($\kappa = 1\%$)	31.18 ±1.53	57.36 ±2.54	27.90 ±1.35	33.67 ±1.79	43.86 ±2.07	60.77 ±6.91
	SO ($\kappa = 2\%$)	31.38 ±1.48	57.82 ±2.54	27.29 ±2.03	34.73 ±1.80	44.13 ±2.37	60.96 ±6.95
	SO ($\kappa = 5\%$)	31.69 ±1.16	58.36 ±2.57	25.17 ±1.70	35.60 ±2.16	44.76 ±2.73	60.58 ±7.93
	SO ($\kappa = 8\%$)	32.22 ±1.15	58.80 ±2.68	25.36 ±3.25	35.78 ±1.66	44.64 ±2.10	61.99 ±7.72
	SO ($\kappa = 10\%$)	31.58 ±1.45	58.87 ±2.33	24.39 ±1.71	36.01 ±1.48	45.02 ±2.29	60.77 ±7.75
	Adam	32.46 ±1.66	59.60 ±2.70	22.22 ±2.32	38.46 ±2.92	46.21 ±3.46	59.55 ±9.51
8	LoRA ($r = 2$)	36.70 ±1.02	59.86 ±1.70	26.62 ±2.47	37.21 ±1.76	51.45 ±3.25	65.45 ±6.55
	LoRA ($r = 4$)	37.90 ±0.92	61.61 ±1.44	28.55 ±2.24	38.25 ±2.02	52.87 ±3.43	65.45 ±5.82
	LoRA ($r = 8$)	38.38 ±1.16	62.03 ±1.03	28.76 ±3.17	39.52 ±2.21	53.04 ±2.82	64.29 ±7.44
	LoRA ($r = 16$)	38.62 ±0.85	61.72 ±1.15	28.04 ±2.79	39.69 ±1.99	53.04 ±3.26	65.00 ±8.04
	ReLoRA ($r = 2$)	36.61 ±1.08	59.86 ±1.70	26.62 ±2.47	37.21 ±1.76	50.83 ±3.77	64.74 ±77.8
	ReLoRA ($r = 4$)	37.77 ±0.95	61.61 ±1.44	28.55 ±2.24	38.25 ±2.02	52.20 ±3.79	65.00 ±6.16
	ReLoRA ($r = 8$)	38.20 ±1.42	62.03 ±1.03	28.76 ±3.17	39.52 ±2.21	53.02 ±3.25	65.51 ±7.64
	ReLoRA ($r = 16$)	38.58 ±0.86	61.72 ±1.15	28.04 ±2.79	39.69 ±1.99	53.10 ±3.50	64.55 ±5.89
	GaLoRE ($r = 2$)	38.91 ±1.04	62.36 ±1.48	27.24 ±2.50	39.02 ±1.87	51.22 ±3.20	63.59 ±7.91
	GaLoRE ($r = 4$)	39.57 ±0.90	63.60 ±1.51	25.09 ±2.54	40.06 ±1.13	51.52 ±2.39	64.94 ±5.73
	GaLoRE ($r = 8$)	39.79 ±1.08	64.06 ±0.98	24.99 ±2.38	40.73 ±1.40	49.89 ±2.75	62.76 ±8.85
	GaLoRE ($r = 16$)	41.42 ±1.00	65.25 ±1.85	24.29 ±2.17	41.08 ±2.34	49.72 ±2.54	63.97 ±5.27
	SO ($\kappa = 1\%$)	40.11 ±1.09	62.62 ±1.06	30.10 ±2.60	39.62 ±1.27	51.08 ±2.88	63.27 ±5.68
	SO ($\kappa = 2\%$)	40.75 ±0.98	63.29 ±1.15	29.57 ±2.19	40.45 ±1.53	51.10 ±2.93	63.78 ±6.62
	SO ($\kappa = 5\%$)	41.14 ±0.97	63.71 ±1.19	27.79 ±2.47	40.88 ±1.20	51.45 ±3.05	63.91 ±8.36
	SO ($\kappa = 8\%$)	41.17 ±1.12	64.71 ±1.18	26.80 ±2.24	41.09 ±1.23	51.76 ±3.13	64.87 ±7.45
	SO ($\kappa = 10\%$)	41.39 ±1.01	64.26 ±0.96	26.46 ±2.49	41.57 ±1.04	51.65 ±3.46	63.53 ±7.02
	Adam	42.39 ±1.26	65.49 ±1.36	23.83 ±2.34	43.25 ±1.59	52.95 ±3.11	64.49 ±8.93
16	LoRA ($r = 2$)	45.15 ±0.98	65.73 ±3.34	29.11 ±2.49	40.56 ±2.89	54.11 ±1.62	62.50 ±9.17
	LoRA ($r = 4$)	46.46 ±0.67	67.36 ±1.11	28.90 ±2.48	44.60 ±1.73	55.28 ±2.37	61.99 ±7.44
	LoRA ($r = 8$)	47.49 ±0.96	68.14 ±0.97	29.59 ±1.50	46.38 ±2.12	56.46 ±2.85	61.92 ±8.63
	LoRA ($r = 16$)	48.07 ±1.01	68.63 ±1.27	29.75 ±1.99	45.68 ±1.57	55.29 ±2.53	63.01 ±9.41
	ReLoRA ($r = 2$)	44.91 ±0.99	65.73 ±3.34	29.11 ±2.49	40.56 ±2.89	53.88 ±1.87	62.50 ±9.17
	ReLoRA ($r = 4$)	46.38 ±0.63	67.36 ±1.11	28.90 ±2.48	44.60 ±1.73	55.59 ±2.13	61.86 ±8.34
	ReLoRA ($r = 8$)	47.49 ±0.99	68.29 ±1.36	29.67 ±2.42	46.40 ±1.81	55.14 ±2.13	61.86 ±8.34
	ReLoRA ($r = 16$)	47.68 ±0.94	68.63 ±1.27	29.75 ±1.99	45.68 ±1.57	55.77 ±2.21	61.99 ±8.00
	GaLoRE ($r = 2$)	47.15 ±0.88	68.05 ±1.33	28.63 ±2.87	46.52 ±2.33	55.61 ±2.39	61.92 ±8.58
	GaLoRE ($r = 4$)	48.13 ±1.13	69.47 ±1.52	26.63 ±1.83	46.75 ±1.75	55.13 ±2.30	61.41 ±8.83
	GaLoRE ($r = 8$)	48.47 ±1.03	69.97 ±1.83	26.26 ±1.63	47.41 ±1.64	54.57 ±1.23	61.22 ±10.40
	GaLoRE ($r = 16$)	50.30 ±0.67	70.36 ±1.59	26.73 ±1.57	47.96 ±1.78	54.42 ±1.41	62.05 ±8.01
	SO ($\kappa = 1\%$)	49.33 ±0.76	68.91 ±1.63	32.79 ±1.99	47.86 ±1.65	55.70 ±1.31	61.28 ±10.38
	SO ($\kappa = 2\%$)	49.95 ±0.68	69.55 ±1.43	31.72 ±2.13	48.09 ±2.05	56.01 ±1.14	62.88 ±7.74
	SO ($\kappa = 5\%$)	50.59 ±0.64	69.92 ±1.66	30.21 ±2.25	48.27 ±1.99	56.16 ±1.47	61.92 ±8.78
	SO ($\kappa = 8\%$)	50.55 ±0.81	70.16 ±1.77	29.34 ±1.57	49.42 ±1.78	56.41 ±1.31	61.03 ±10.07
	SO ($\kappa = 10\%$)	50.68 ±0.56	70.30 ±1.66	28.04 ±1.87	49.23 ±1.67	56.41 ±1.19	62.18 ±10.51
	Adam	51.12 ±0.84	70.51 ±1.69	25.94 ±1.28	50.09 ±1.59	56.50 ±1.97	61.99 ±10.34

Table 19. Few-shot classification performance on 6 datasets using a two-layer fully-connected architecture after pretraining on FMNIST. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Shots	**Model**	**EMNIST**	**MNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
4	LoRA ($r = 2$)	29.31 ±1.43	54.60 ±3.53	28.00 ±2.63	35.65 ±2.42	40.59 ±2.68	60.38 ±9.03
	LoRA ($r = 4$)	29.55 ±1.19	54.73 ±2.93	28.04 ±2.62	36.13 ±1.63	42.27 ±2.08	57.88 ±9.11
	LoRA ($r = 8$)	30.01 ±1.68	54.84 ±3.41	28.61 ±2.12	36.43 ±1.57	42.19 ±3.11	57.76 ±8.97
	LoRA ($r = 16$)	30.21 ±1.46	55.64 ±3.45	28.85 ±1.96	36.03 ±1.61	41.74 ±2.83	58.46 ±9.96
	ReLoRA ($r = 2$)	29.32 ±1.43	54.60 ±3.53	28.00 ±2.63	35.65 ±2.42	40.59 ±2.68	60.38 ±9.03
	ReLoRA ($r = 4$)	29.46 ±1.32	54.73 ±2.93	28.04 ±2.62	36.13 ±1.63	42.27 ±2.08	57.88 ±9.11
	ReLoRA ($r = 8$)	29.85 ±1.64	54.84 ±3.41	28.61 ±2.12	36.43 ±1.57	42.19 ±3.11	57.76 ±8.97
	ReLoRA ($r = 16$)	30.06 ±1.36	55.64 ±3.45	28.85 ±1.96	36.03 ±1.61	41.74 ±2.83	58.46 ±9.96
	GaLoRE ($r = 2$)	29.87 ±1.27	55.90 ±3.22	26.72 ±2.58	35.84 ±1.58	40.77 ±2.71	58.27 ±9.02
	GaLoRE ($r = 4$)	30.62 ±1.43	56.53 ±3.81	25.57 ±1.98	36.54 ±2.43	43.23 ±2.98	58.72 ±8.83
	GaLoRE ($r = 8$)	31.45 ±1.17	58.17 ±3.36	24.33 ±2.09	35.69 ±1.59	41.97 ±2.24	56.54 ±9.39
	GaLoRE ($r = 16$)	32.14 ±1.07	58.80 ±2.70	23.33 ±1.85	36.06 ±1.25	41.51 ±2.76	57.69 ±7.43
	SO ($\kappa = 1\%$)	31.05 ±1.01	55.36 ±2.85	27.46 ±2.12	35.61 ±1.47	39.69 ±2.40	57.44 ±7.47
	SO ($\kappa = 2\%$)	31.47 ±1.22	56.00 ±3.43	26.97 ±2.03	36.23 ±1.46	40.68 ±2.43	57.12 ±7.61
	SO ($\kappa = 5\%$)	32.46 ±1.14	57.86 ±2.49	25.76 ±1.88	36.34 ±1.57	41.65 ±1.89	57.88 ±8.14
	SO ($\kappa = 8\%$)	32.67 ±1.24	57.55 ±2.98	25.53 ±2.17	37.00 ±2.29	41.86 ±2.43	57.12 ±7.66
	SO ($\kappa = 10\%$)	32.90 ±1.22	59.12 ±3.30	25.32 ±1.42	36.86 ±1.63	42.25 ±2.30	57.88 ±7.52
	Adam	33.11 ±1.54	60.18 ±2.96	22.58 ±2.52	39.26 ±1.65	44.36 ±2.00	56.73 ±7.09
8	LoRA ($r = 2$)	37.38 ±0.95	65.74 ±2.44	28.57 ±2.33	41.26 ±1.45	48.66 ±3.46	59.62 ±5.64
	LoRA ($r = 4$)	38.57 ±1.00	66.13 ±3.10	28.93 ±2.57	42.89 ±1.70	49.46 ±3.70	60.26 ±7.22
	LoRA ($r = 8$)	38.87 ±0.89	67.14 ±3.31	29.71 ±2.26	43.20 ±1.49	48.68 ±3.83	60.90 ±5.47
	LoRA ($r = 16$)	39.03 ±1.08	66.97 ±2.66	29.43 ±1.91	43.07 ±1.93	49.41 ±4.15	60.51 ±6.78
	ReLoRA ($r = 2$)	37.15 ±1.08	65.74 ±2.44	28.57 ±2.33	41.26 ±1.45	48.66 ±3.46	59.62 ±5.64
	ReLoRA ($r = 4$)	38.38 ±0.92	66.13 ±3.10	28.93 ±2.57	42.89 ±1.70	49.46 ±3.70	60.26 ±7.22
	ReLoRA ($r = 8$)	38.79 ±0.92	67.14 ±3.31	29.71 ±2.26	43.20 ±1.49	48.68 ±3.83	60.90 ±5.47
	ReLoRA ($r = 16$)	39.13 ±1.14	66.97 ±2.66	29.43 ±1.91	43.07 ±1.93	49.41 ±4.15	60.51 ±6.78
	GaLoRE ($r = 2$)	38.47 ±1.05	66.38 ±3.41	27.81 ±2.58	41.43 ±156	47.57 ±3.39	61.28 ±5.78
	GaLoRE ($r = 4$)	39.19 ±0.81	68.95 ±2.89	26.10 ±2.72	42.64 ±1.41	48.31 ±3.58	60.90 ±5.81
	GaLoRE ($r = 8$)	40.41 ±1.14	69.20 ±2.94	24.94 ±2.27	42.00 ±1.81	46.92 ±3.58	60.58 ±6.43
	GaLoRE ($r = 16$)	41.93 ±1.28	70.05 ±2.69	24.61 ±2.34	42.89 ±1.02	47.42 ±3.30	62.24 ±6.06
	SO ($\kappa = 1\%$)	40.63 ±0.91	67.70 ±3.16	29.45 ±1.99	42.27 ±1.38	45.77 ±3.55	58.78 ±6.46
	SO ($\kappa = 2\%$)	41.29 ±1.07	68.03 ±3.09	28.80 ±2.54	42.52 ±1.29	46.74 ±3.32	58.21 ±6.67
	SO ($\kappa = 5\%$)	41.95 ±0.89	69.18 ±2.98	27.42 ±2.42	42.92 ±1.69	47.33 ±2.89	60.00 ±6.49
	SO ($\kappa = 8\%$)	42.09 ±0.95	69.60 ±3.03	26.97 ±2.90	43.53 ±1.30	47.70 ±3.51	60.96 ±7.88
	SO ($\kappa = 10\%$)	42.08 ±1.09	69.60 ±3.09	26.36 ±2.68	44.15 ±1.67	48.85 ±3.58	61.09 ±6.93
	Adam	42.84 ±1.07	70.67 ±2.38	23.97 ±2.88	44.76 ±1.90	50.44 ±4.18	62.44 ±6.04
16	LoRA ($r = 2$)	44.36 ±0.64	72.90 ±2.01	29.86 ±3.08	46.84 ±2.30	54.73 ±2.34	61.09 ±9.21
	LoRA ($r = 4$)	45.96 ±0.49	75.26 ±1.59	30.26 ±3.06	48.21 ±1.26	55.55 ±2.24	61.47 ±9.88
	LoRA ($r = 8$)	47.19 ±0.45	75.91 ±1.15	30.91 ±3.03	48.25 ±1.40	55.47 ±2.16	61.35 ±9.27
	LoRA ($r = 16$)	47.80 ±0.61	75.58 ±1.44	31.16 ±2.73	48.93 ±2.06	55.22 ±2.20	62.31 ±8.76
	ReLoRA ($r = 2$)	44.51 ±0.65	72.90 ±2.01	29.86 ±30.8	46.84 ±2.30	54.73 ±2.34	61.09 ±9.21
	ReLoRA ($r = 4$)	46.14 ±0.65	75.26 ±1.59	30.26 ±3.06	48.21 ±1.26	55.55 ±2.24	61.47 ±9.88
	ReLoRA ($r = 8$)	47.20 ±0.50	75.91 ±1.15	30.91 ±3.03	48.25 ±1.40	55.47 ±2.16	61.35 ±9.27
	ReLoRA ($r = 16$)	47.64 ±0.43	75.74 ±1.68	31.30 ±2.72	48.67 ±2.30	55.37 ±2.11	60.71 ±8.95
	GaLoRE ($r = 2$)	46.45 ±0.53	75.22 ±1.11	29.86 ±1.63	47.88 ±2.01	53.39 ±1.52	61.03 ±9.89
	GaLoRE ($r = 4$)	47.46 ±0.54	75.74 ±1.50	27.47 ±1.73	47.64 ±2.11	53.33 ±1.94	60.90 ±8.08
	GaLoRE ($r = 8$)	48.43 ±0.72	77.44 ±1.33	26.85 ±1.96	49.06 ±1.94	52.53 ±2.37	61.22 ±9.43
	GaLoRE ($r = 16$)	49.87 ±0.86	78.36 ±1.45	26.64 ±1.41	49.30 ±1.54	51.87 ±1.91	59.29 ±9.01
	SO ($\kappa = 1\%$)	49.54 ±0.51	76.59 ±1.36	32.65 ±1.72	48.88 ±1.85	52.08 ±1.42	60.38 ±8.87
	SO ($\kappa = 2\%$)	49.62 ±0.60	77.41 ±1.38	31.35 ±1.19	49.17 ±1.33	52.91 ±1.53	59.23 ±9.47
	SO ($\kappa = 5\%$)	50.36 ±0.60	78.14 ±1.42	29.88 ±1.75	49.25 ±1.74	53.62 ±1.72	60.64 ±9.06
	SO ($\kappa = 8\%$)	50.06 ±0.98	78.25 ±1.65	29.38 ±2.01	50.01 ±1.50	53.81 ±1.18	59.36 ±9.51
	SO ($\kappa = 10\%$)	50.21 ±0.90	78.83 ±1.32	27.86 ±2.20	49.66 ±1.55	54.79 ±1.43	60.71 ±8.31
	Adam	51.49 ±0.50	78.8 ±1.42	26.28 ±1.67	50.84 ±1.47	56.02 ±1.71	60.83 ±8.77

Table 20. Few-shot classification performance on 7 datasets using a two-layer fully-connected architecture without pretraining. Results are the average top-1 accuracy of 10 executions ± standard deviation.

Shots	**Strategy**	**EMNIST**	**MNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
4	**Importance-Based Gradient Pruning**							
	SO ($\kappa = 1\%$)	26.59 ±2.77	58.43 ±3.20	58.71 ±2.69	23.85 ±4.78	38.12 ±2.13	46.15 ±3.25	59.23 ±8.68
	SO ($\kappa = 2\%$)	29.31 ±1.76	59.94 ±4.18	59.44 ±2.70	23.83 ±4.50	40.27 ±1.80	46.07 ±2.05	57.50 ±8.26
	SO ($\kappa = 5\%$)	30.44 ±3.35	61.16 ±4.29	61.32 ±2.23	22.77 ±4.89	39.86 ±1.73	46.16 ±3.43	56.99 ±9.70
	SO ($\kappa = 8\%$)	31.97 ±1.52	61.50 ±3.76	60.97 ±2.91	23.39 ±5.31	40.32 ±1.29	47.21 ±2.92	57.63 ±9.66
	SO ($\kappa = 10\%$)	32.23 ±1.64	61.80 ±2.85	61.26 ±2.49	21.74 ±3.60	39.92 ±1.50	46.61 ±3.39	56.73 ±9.65
	Random Gradient Pruning							
	SO ($\kappa = 1\%$)	34.17 ±1.83	63.20 ±3.03	61.15 ±2.78	26.61 ±5.09	40.72 ±1.79	46.47 ±2.64	57.88 ±11.01
	SO ($\kappa = 2\%$)	34.38 ±1.81	63.14 ±3.10	61.32 ±2.79	25.77 ±4.08	41.25 ±2.33	45.94 ±2.47	57.12 ±10.52
	SO ($\kappa = 5\%$)	34.07 ±1.84	62.98 ±2.52	61.81 ±2.21	22.93 ±4.88	40.92 ±2.37	46.94 ±2.50	56.86 ±10.26
	SO ($\kappa = 8\%$)	33.93 ±1.73	63.59 ±3.10	61.60 ±2.40	24.30 ±4.84	40.87 ±1.91	46.74 ±2.51	57.12 ±9.87
	SO ($\kappa = 10\%$)	34.22 ±1.80	63.02 ±3.32	61.16 ±2.22	21.27 ±2.96	40.08 ±1.87	45.42 ±2.24	58.46 ±10.54
8	**Importance-Based Gradient Pruning**							
	SO ($\kappa = 1\%$)	32.19 ±3.40	68.11 ±2.40	64.99 ±2.74	23.66 ±3.92	42.76 ±1.86	51.27 ±3.43	62.05 ±9.75
	SO ($\kappa = 2\%$)	37.57 ±1.50	69.88 ±2.58	66.51 ±1.42	23.39 ±4.34	44.58 ±2.43	52.05 ±3.54	63.27 ±8.28
	SO ($\kappa = 5\%$)	39.54 ±1.51	72.18 ±2.17	67.16 ±1.22	21.83 ±3.09	45.20 ±1.21	52.26 ±3.56	63.65 ±8.22
	SO ($\kappa = 8\%$)	40.28 ±1.31	72.32 ±2.23	66.99 ±1.01	23.33 ±4.62	45.39 ±2.13	52.34 ±2.77	61.99 ±8.79
	SO ($\kappa = 10\%$)	39.18 ±3.86	72.21 ±2.36	67.12 ±1.51	21.52 ±3.39	45.49 ±1.54	52.46 ±3.48	61.67 ±7.70
	Random Gradient Pruning							
	SO ($\kappa = 1\%$)	41.79 ±1.45	72.09 ±2.36	67.18 ±1.22	26.79 ±4.63	46.65 ±1.47	51.66 ±3.33	64.10 ±8.19
	SO ($\kappa = 2\%$)	41.78 ±1.27	72.34 ±2.48	67.42 ±1.32	25.39 ±3.78	46.66 ±2.43	51.66 ±2.96	63.46 ±7.57
	SO ($\kappa = 5\%$)	41.81 ±1.37	72.22 ±2.72	67.20 ±1.38	24.08 ±3.75	46.36 ±1.72	52.03 ±3.40	63.33 ±8.64
	SO ($\kappa = 8\%$)	41.69 ±1.19	71.86 ±2.47	67.11 ±1.24	23.33 ±3.63	45.87 ±1.87	52.16 ±3.60	63.59 ±8.34
	SO ($\kappa = 10\%$)	41.31 ±1.58	72.52 ±2.37	66.89 ±1.74	23.14 ±2.56	45.98 ±1.99	51.98 ±3.25	63.27 ±7.51
16	**Importance-Based Gradient Pruning**							
	SO ($\kappa = 1\%$)	41.94 ±1.16	75.87 ±1.70	69.98 ±1.55	24.26 ±2.62	47.76 ±1.47	55.59 ±2.12	60.71 ±9.17
	SO ($\kappa = 2\%$)	43.02 ±4.11	77.35 ±1.80	70.69 ±1.02	22.54 ±1.75	49.28 ±1.87	56.21 ±1.55	60.19 ±8.50
	SO ($\kappa = 5\%$)	44.88 ±4.88	78.63 ±1.54	71.41 ±1.01	23.83 ±1.87	50.19 ±1.41	56.20 ±1.99	61.54 ±9.50
	SO ($\kappa = 8\%$)	45.36 ±4.66	78.69 ±1.43	71.20 ±1.20	23.55 ±1.77	50.68 ±1.23	56.78 ±2.35	61.47 ±7.88
	SO ($\kappa = 10\%$)	46.63 ±3.50	78.84 ±1.45	71.58 ±1.24	23.24 ±2.35	51.19 ±1.72	57.10 ±1.88	60.13 ±9.21
	Random Gradient Pruning							
	SO ($\kappa = 1\%$)	48.60 ±0.68	78.99 ±1.39	71.94 ±1.57	27.21 ±2.55	52.85 ±1.19	55.93 ±1.68	62.12 ±9.50
	SO ($\kappa = 2\%$)	48.38 ±0.49	78.64 ±1.66	71.90 ±1.24	26.77 ±2.35	52.15 ±1.43	55.51 ±2.19	62.69 ±8.24
	SO ($\kappa = 5\%$)	48.45 ±0.82	79.45 ±1.46	71.44 ±1.66	27.06 ±2.65	51.77 ±1.62	56.38 ±1.80	61.15 ±9.45
	SO ($\kappa = 8\%$)	48.74 ±0.71	79.17 ±1.58	71.79 ±1.43	25.25 ±1.84	51.71 ±1.12	56.71 ±1.81	63.97 ±8.25
	SO ($\kappa = 10\%$)	48.77 ±1.07	79.06 ±1.53	71.90 ±1.34	25.50 ±2.39	51.77 ±1.19	56.81 ±1.58	60.51 ±10.17

Table 21. Few-shot classification performance on 6 datasets using a two-layer fully-connected architecture after pretraining on MNIST. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Shots	**Strategy**	**EMNIST**	**FMNIST**	**PathMNIST**	**OrganMNISTAxial**	**BloodMNIST**	**BreastMNIST**
4	**Importance-Based Gradient Pruning**						
	SO ($\kappa = 1\%$)	30.81 ±1.75	59.46 ±2.45	24.87 ±3.73	35.45 ±2.31	44.15 ±3.04	60.71 ±7.35
	SO ($\kappa = 2\%$)	31.25 ±1.25	59.37 ±2.65	23.00 ±2.96	36.52 ±1.83	44.72 ±3.01	61.03 ±8.59
	SO ($\kappa = 5\%$)	32.17 ±1.69	59.44 ±2.54	22.84 ±3.37	36.91 ±1.56	44.40 ±3.12	60.13 ±7.87
	SO ($\kappa = 8\%$)	32.02 ±2.42	60.23 ±2.18	23.01 ±3.31	36.54 ±1.94	44.84 ±1.62	61.28 ±6.62
	SO ($\kappa = 10\%$)	31.72 ±1.46	60.12 ±2.51	22.77 ±2.92	36.12 ±2.00	44.60 ±2.65	60.26 ±9.27
	Random Gradient Pruning						
	SO ($\kappa = 1\%$)	31.18 ±1.53	57.36 ±2.54	27.90 ±1.35	33.67 ±1.79	43.86 ±2.07	60.77 ±6.91
	SO ($\kappa = 2\%$)	31.38 ±1.48	57.82 ±2.54	27.29 ±2.03	34.73 ±1.80	44.13 ±2.37	60.96 ±6.95
	SO ($\kappa = 5\%$)	31.69 ±1.16	58.36 ±2.57	25.17 ±1.70	35.60 ±2.16	44.76 ±2.73	60.58 ±7.93
	SO ($\kappa = 8\%$)	32.22 ±1.15	58.80 ±2.68	25.36 ±3.25	35.78 ±1.66	44.64 ±2.10	61.99 ±7.72
	SO ($\kappa = 10\%$)	31.58 ±1.45	58.87 ±2.33	24.39 ±1.71	36.01 ±1.48	45.02 ±2.29	60.77 ±7.75
8	**Importance-Based Gradient Pruning**						
	SO ($\kappa = 1\%$)	39.95 ±1.05	64.84 ±1.37	25.48 ±3.08	41.26 ±2.03	51.78 ±2.86	63.27 ±7.98
	SO ($\kappa = 2\%$)	40.75 ±0.98	65.27 ±1.57	24.56 ±3.18	40.95 ±1.58	51.52 ±3.13	63.33 ±7.27
	SO ($\kappa = 5\%$)	40.56 ±2.61	65.26 ±1.23	23.03 ±3.33	41.51 ±1.33	51.09 ±3.33	63.53 ±7.27
	SO ($\kappa = 8\%$)	41.60 ±0.91	65.49 ±1.29	23.22 ±2.82	42.22 ±1.81	51.21 ±2.46	63.91 ±7.80
	SO ($\kappa = 10\%$)	40.51 ±2.64	65.70 ±1.50	23.04 ±3.13	41.68 ±1.30	50.64 ±4.60	63.72 ±7.15
	Random Gradient Pruning						
	SO ($\kappa = 1\%$)	40.11 ±1.09	62.62 ±1.06	30.10 ±2.60	39.62 ±1.27	51.08 ±2.88	63.27 ±5.68
	SO ($\kappa = 2\%$)	40.75 ±0.98	63.29 ±1.15	29.57 ±2.19	40.45 ±1.53	51.10 ±2.93	63.78 ±6.62
	SO ($\kappa = 5\%$)	41.14 ±0.97	63.71 ±1.19	27.79 ±2.47	40.88 ±1.20	51.45 ±3.05	63.91 ±8.36
	SO ($\kappa = 8\%$)	41.17 ±1.12	64.71 ±1.18	26.80 ±2.24	41.09 ±1.23	51.76 ±3.13	64.87 ±7.45
	SO ($\kappa = 10\%$)	41.39 ±1.01	64.26 ±0.96	26.46 ±2.49	41.57 ±1.04	51.65 ±3.46	63.53 ±7.02
16	**Importance-Based Gradient Pruning**						
	SO ($\kappa = 1\%$)	47.81 ±1.56	70.38 ±1.43	27.52 ±2.30	47.18 ±1.83	55.51 ±1.98	60.58 ±9.49
	SO ($\kappa = 2\%$)	48.19 ±1.12	70.60 ±1.32	26.73 ±2.46	48.50 ±1.75	55.62 ±1.89	60.71 ±7.15
	SO ($\kappa = 5\%$)	49.99 ±0.81	70.55 ±1.19	26.21 ±2.48	48.67 ±2.55	55.39 ±1.70	60.83 ±8.18
	SO ($\kappa = 8\%$)	48.68 ±3.76	71.07 ±1.55	25.14 ±2.53	49.12 ±2.02	56.01 ±1.85	60.96 ±9.06
	SO ($\kappa = 10\%$)	48.51 ±4.02	71.24 ±1.28	24.62 ±2.64	49.02 ±1.42	55.79 ±1.71	63.72 ±7.20
	Random Gradient Pruning						
	SO ($\kappa = 1\%$)	49.33 ±0.76	68.91 ±1.63	32.79 ±1.99	47.86 ±1.65	55.70 ±1.31	61.28 ±10.38
	SO ($\kappa = 2\%$)	49.95 ±0.68	69.55 ±1.43	31.72 ±2.13	48.09 ±2.05	56.01 ±1.14	62.88 ±7.74
	SO ($\kappa = 5\%$)	50.59 ±0.64	69.92 ±1.66	30.21 ±2.25	48.27 ±1.99	56.16 ±1.47	61.92 ±8.78
	SO ($\kappa = 8\%$)	50.55 ±0.81	70.16 ±1.77	29.34 ±1.57	49.42 ±1.78	56.41 ±1.31	61.03 ±10.07
	SO ($\kappa = 10\%$)	50.68 ±0.56	70.30 ±1.66	28.04 ±1.87	49.23 ±1.67	56.41 ±1.19	62.18 ±10.51

Table 22. Few-shot classification performance on 6 datasets using a two-layer fully-connected architecture after pretraining on FMNIST. Results are the average top-1 accuracy over 10 executions ± standard deviation.

Shots	Strategy	EMNIST	MNIST	PathMNIST	OrganMNISTAxial	BloodMNIST	BreastMNIST
4	**Importance-Based Gradient Pruning**						
	SO ($\kappa = 1\%$)	31.08 ±0.86	57.27 ±3.66	24.99 ±2.72	36.25 ±1.71	42.15 ±3.21	57.50 ±8.22
	SO ($\kappa = 2\%$)	31.44 ±1.06	58.28 ±3.61	25.26 ±2.91	36.74 ±1.81	42.56 ±3.83	57.12 ±7.89
	SO ($\kappa = 5\%$)	32.12 ±1.28	59.20 ±3.82	23.97 ±2.64	37.71 ±1.49	43.18 ±2.93	58.01 ±7.08
	SO ($\kappa = 8\%$)	31.48 ±2.11	59.55 ±2.78	22.94 ±2.38	37.20 ±2.13	42.36 ±2.86	56.79 ±7.75
	SO ($\kappa = 10\%$)	33.19 ±1.36	58.83 ±3.85	23.68 ±2.32	37.61 ±1.74	42.50 ±2.92	58.21 ±9.28
	Random Gradient Pruning						
	SO ($\kappa = 1\%$)	31.05 ±1.01	55.36 ±2.85	27.46 ±2.12	35.61 ±1.47	39.69 ±2.40	57.44 ±7.47
	SO ($\kappa = 2\%$)	31.47 ±1.22	56.00 ±3.43	26.97 ±2.03	36.23 ±1.46	40.68 ±2.43	57.12 ±7.61
	SO ($\kappa = 5\%$)	32.46 ±1.14	57.86 ±2.49	25.76 ±1.88	36.34 ±1.57	41.65 ±1.89	57.88 ±8.14
	SO ($\kappa = 8\%$)	32.67 ±1.24	57.55 ±2.98	25.53 ±2.17	37.00 ±2.29	41.86 ±2.43	57.12 ±7.66
	SO ($\kappa = 10\%$)	32.90 ±1.22	59.12 ±3.30	25.32 ±1.42	36.86 ±1.63	42.25 ±2.30	57.88 ±7.52
8	**Importance-Based Gradient Pruning**						
	SO ($\kappa = 1\%$)	39.35 ±0.80	69.48 ±3.09	26.29 ±3.45	43.11 ±0.96	48.56 ±3.74	58.53 ±7.07
	SO ($\kappa = 2\%$)	40.43 ±0.87	70.04 ±3.47	25.17 ±3.80	43.81 ±1.92	48.28 ±3.25	62.37 ±5.42
	SO ($\kappa = 5\%$)	41.58 ±1.07	70.74 ±2.67	24.87 ±2.93	44.67 ±1.25	49.41 ±3.84	62.24 ±5.64
	SO ($\kappa = 8\%$)	41.84 ±1.22	71.47 ±2.92	23.81 ±2.85	44.51 ±1.80	48.14 ±4.27	59.42 ±7.51
	SO ($\kappa = 10\%$)	41.12 ±2.07	77.20 ±1.34	24.02 ±3.65	44.34 ±1.52	49.04 ±3.98	62.63 ±8.34
	Random Gradient Pruning						
	SO ($\kappa = 1\%$)	40.63 ±0.91	67.70 ±3.16	29.45 ±1.99	42.27 ±1.38	45.77 ±3.55	58.78 ±6.46
	SO ($\kappa = 2\%$)	41.29 ±1.07	68.03 ±3.09	28.80 ±2.54	42.52 ±1.29	46.74 ±3.32	58.21 ±6.67
	SO ($\kappa = 5\%$)	41.95 ±0.89	69.18 ±2.98	27.42 ±2.42	42.92 ±1.69	47.33 ±2.89	60.00 ±6.49
	SO ($\kappa = 8\%$)	42.09 ±0.95	69.60 ±3.03	26.97 ±2.90	43.53 ±1.30	47.70 ±3.51	60.96 ±7.88
	SO ($\kappa = 10\%$)	42.08 ±1.09	69.60 ±3.09	26.36 ±2.68	44.15 ±1.67	48.85 ±3.58	61.09 ±6.93
16	**Importance-Based Gradient Pruning**						
	SO ($\kappa = 1\%$)	47.09 ±0.84	77.47 ±1.42	27.23 ±1.35	48.99 ±1.67	55.20 ±1.95	60.64 ±8.21
	SO ($\kappa = 2\%$)	47.38 ±1.85	78.00 ±1.48	26.75 ±1.96	49.79 ±1.85	55.08 ±1.91	60.26 ±9.99
	SO ($\kappa = 5\%$)	48.87 ±2.36	78.51 ±1.51	25.77 ±1.92	49.23 ±1.72	55.33 ±1.80	60.00 ±9.27
	SO ($\kappa = 8\%$)	48.07 ±3.95	78.96 ±1.03	23.89 ±3.07	50.07 ±1.48	55.06 ±2.25	61.03 ±8.49
	SO ($\kappa = 10\%$)	50.14 ±0.75	78.16 ±3.82	24.68 ±2.11	50.81 ±1.69	54.09 ±3.10	61.92 ±8.95
	Random Gradient Pruning						
	SO ($\kappa = 1\%$)	49.54 ±0.51	76.59 ±1.36	32.65 ±1.72	48.88 ±1.85	52.08 ±1.42	60.38 ±8.87
	SO ($\kappa = 2\%$)	49.62 ±0.60	77.41 ±1.38	31.35 ±1.19	49.17 ±1.33	52.91 ±1.53	59.23 ±9.47
	SO ($\kappa = 5\%$)	50.36 ±0.60	78.14 ±1.42	29.88 ±1.75	49.25 ±1.74	53.62 ±1.72	60.64 ±9.06
	SO ($\kappa = 8\%$)	50.06 ±0.98	78.25 ±1.65	29.38 ±2.01	50.01 ±1.50	53.81 ±1.18	59.36 ±9.51
	SO ($\kappa = 10\%$)	50.21 ±0.90	78.83 ±1.32	27.86 ±2.20	49.66 ±1.55	54.79 ±1.43	60.71 ±8.31

Figure 5. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Gradient Rank Evolution in Few-Shot Learning (4 shots) on EMNIST Dataset.

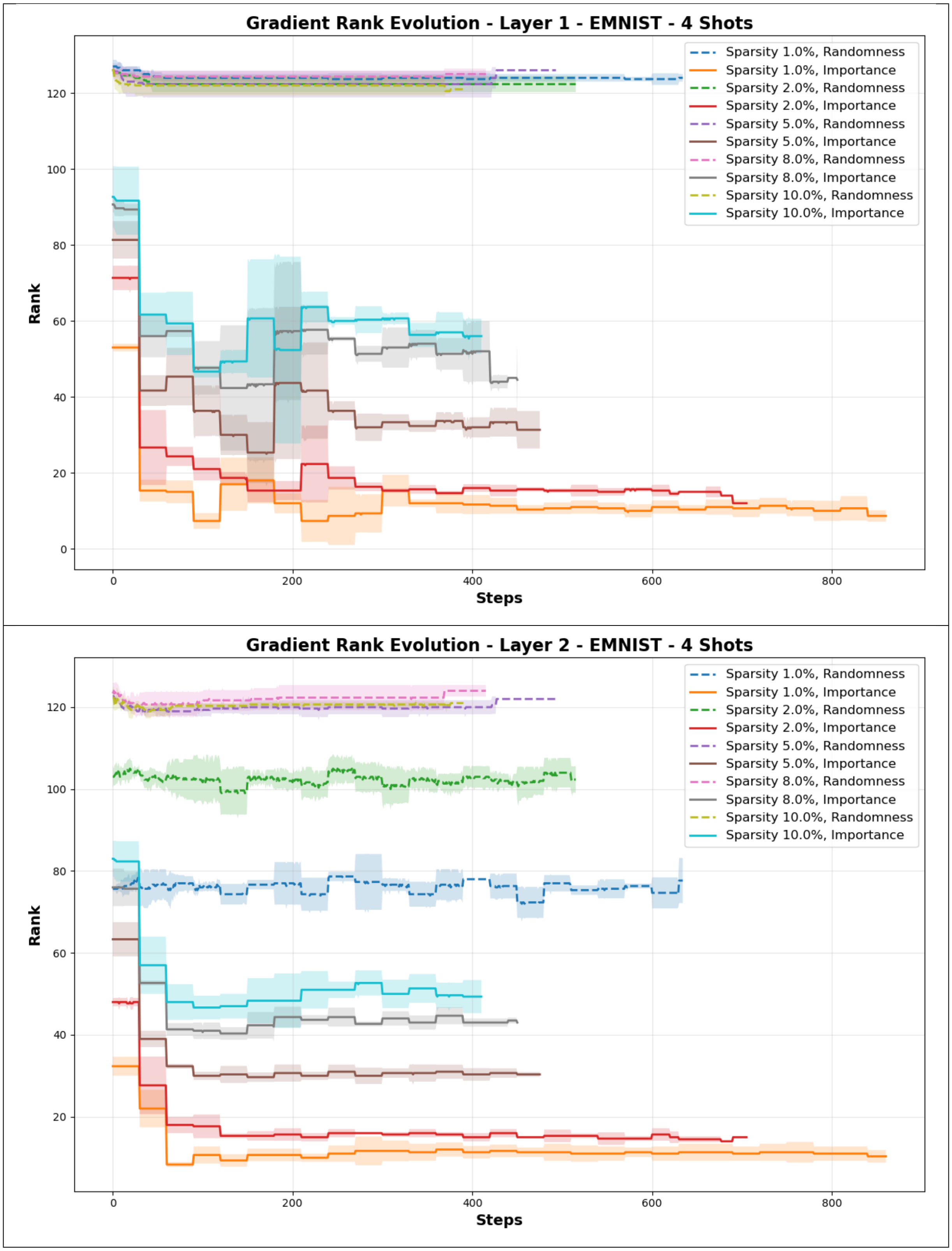


Figure 6. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Loss Evolution in Few-Shot Learning (4 shots) on EMNIST Dataset.

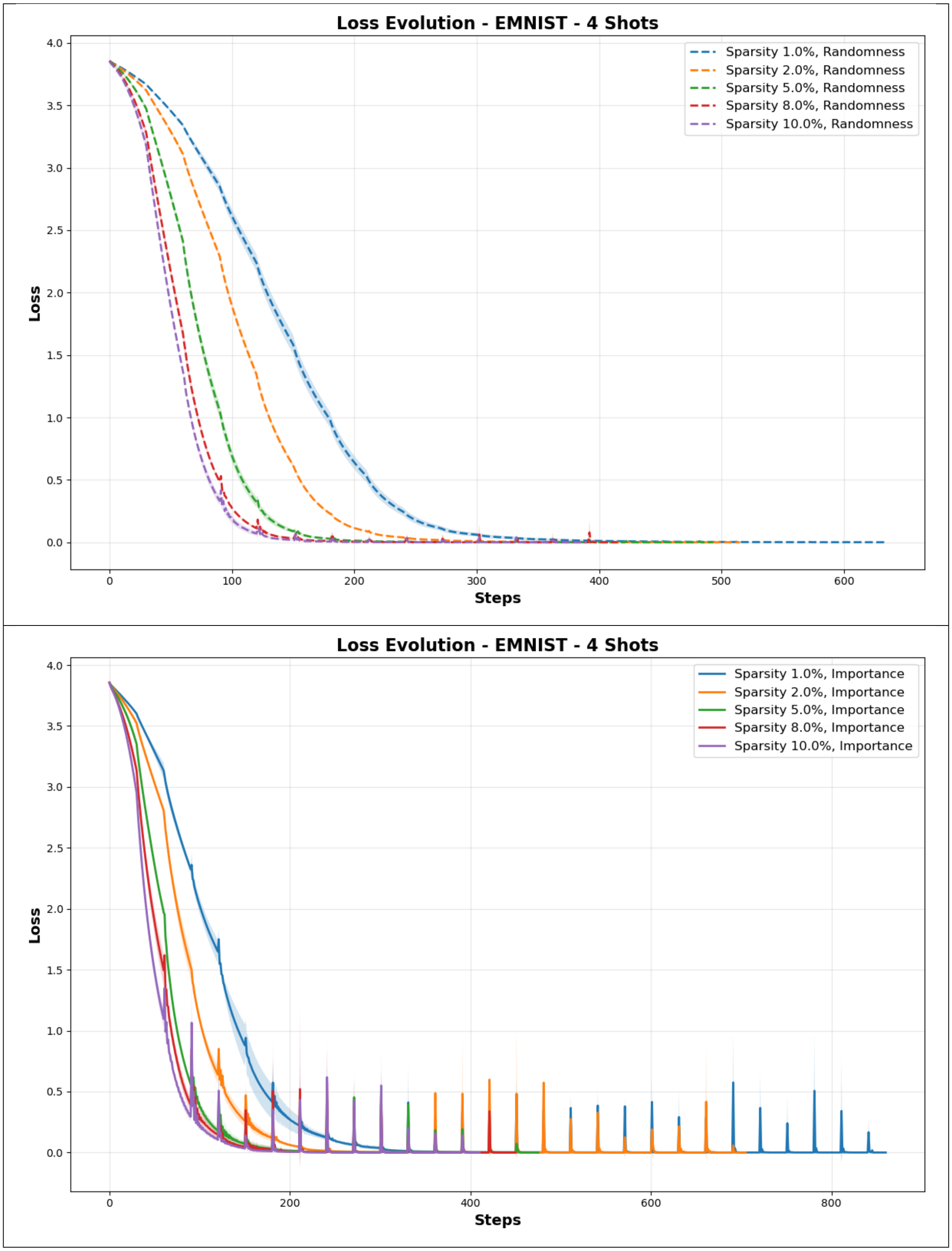


Figure 7. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Gradient Rank Evolution in Few-Shot Learning (8 shots) on EMNIST Dataset.

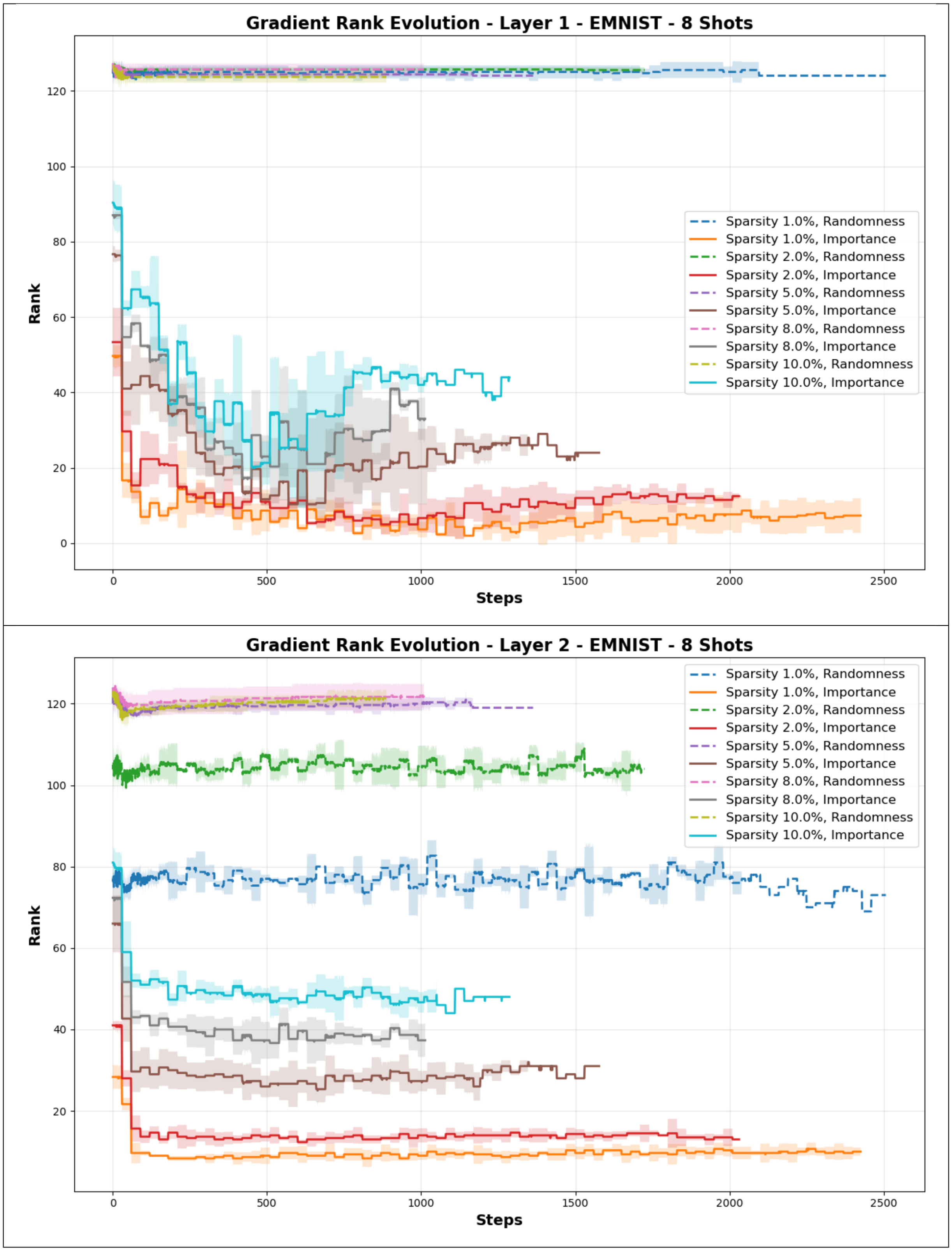


Figure 8. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Loss Evolution of SO in Few-Shot Learning (8 shots) on EMNIST Dataset.

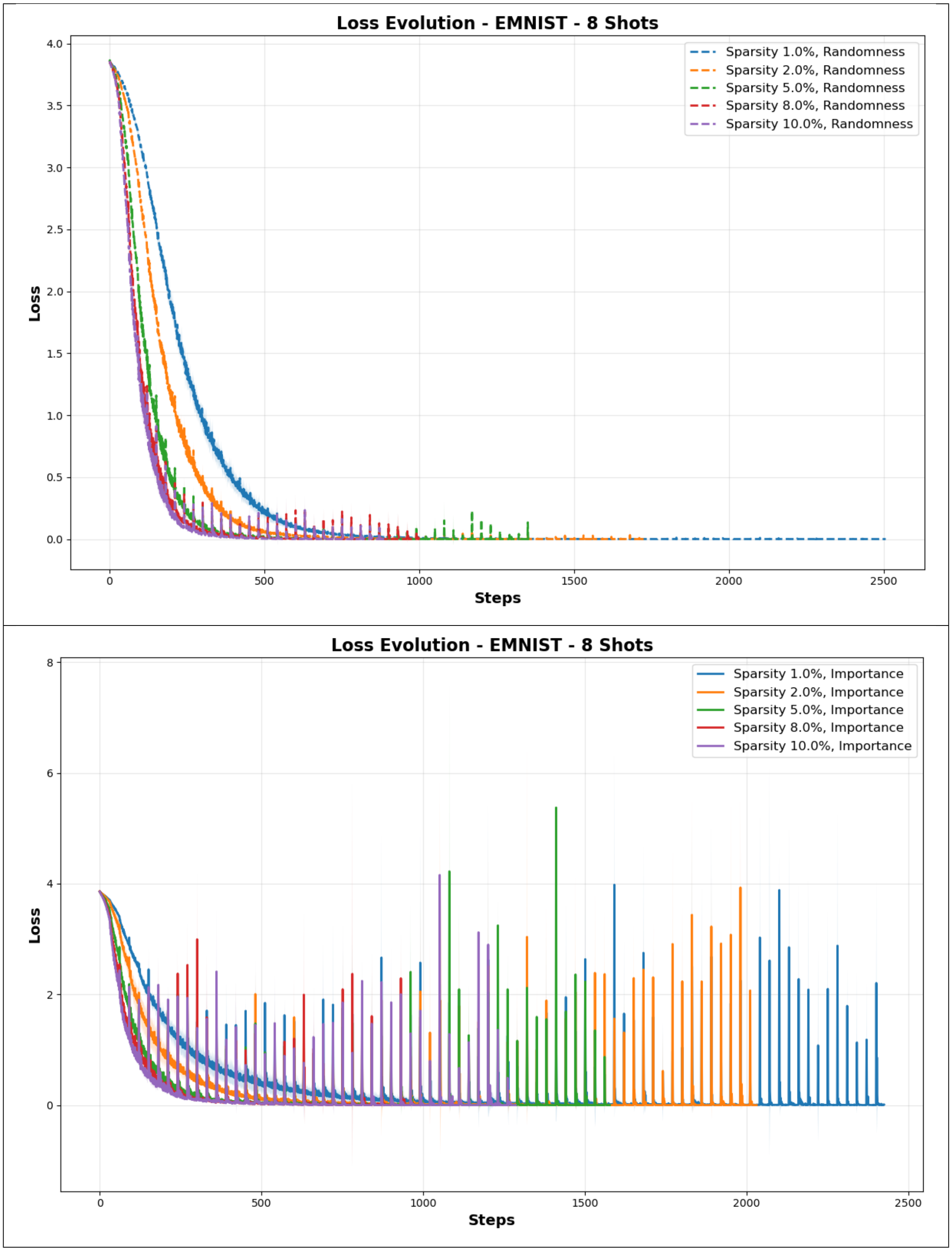


Figure 9. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Gradient Rank Evolution in Few-Shot Learning (16 shots) on EMNIST Dataset.

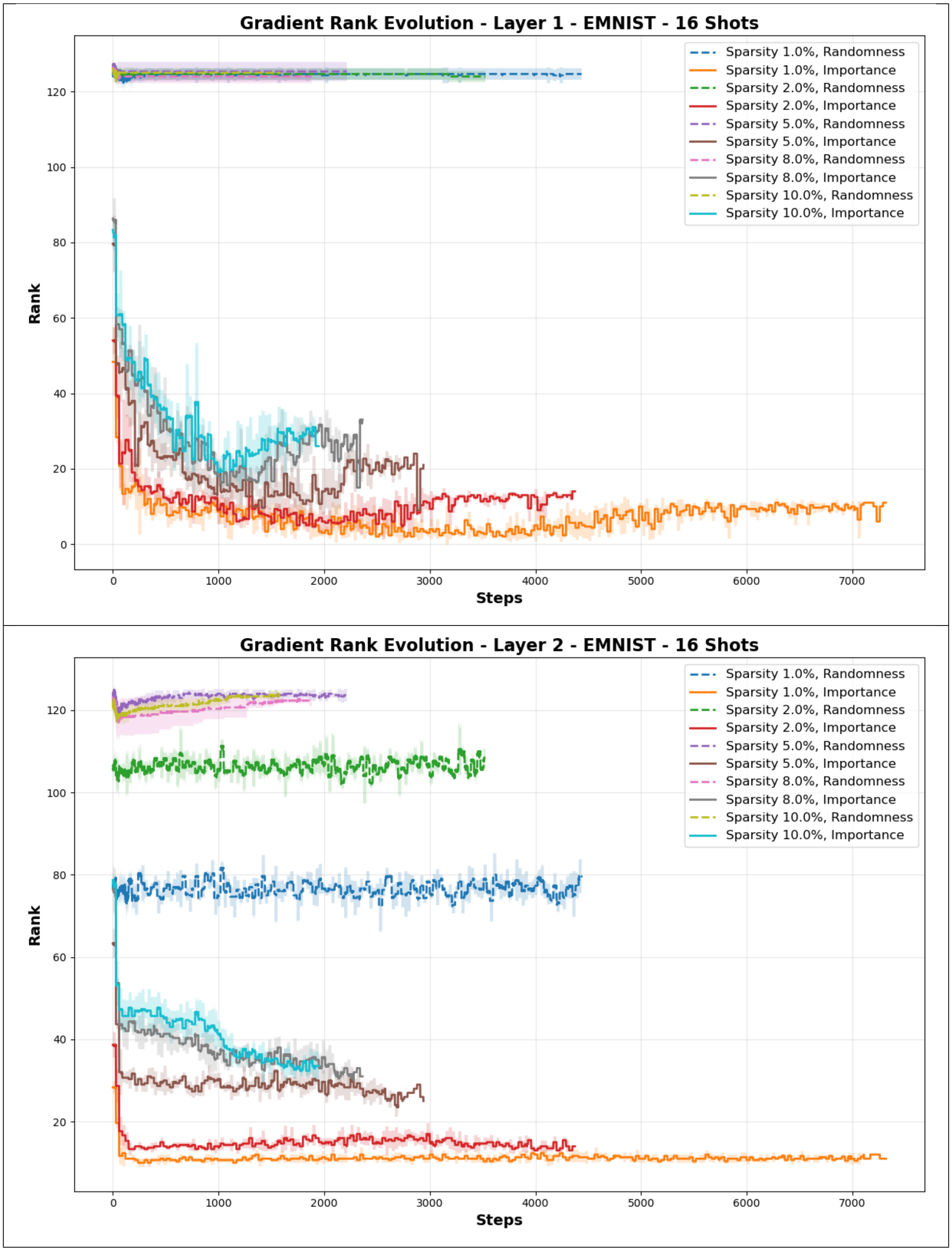


Figure 10. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Loss Evolution in Few-Shot Learning (16 shots) on EMNIST Dataset.

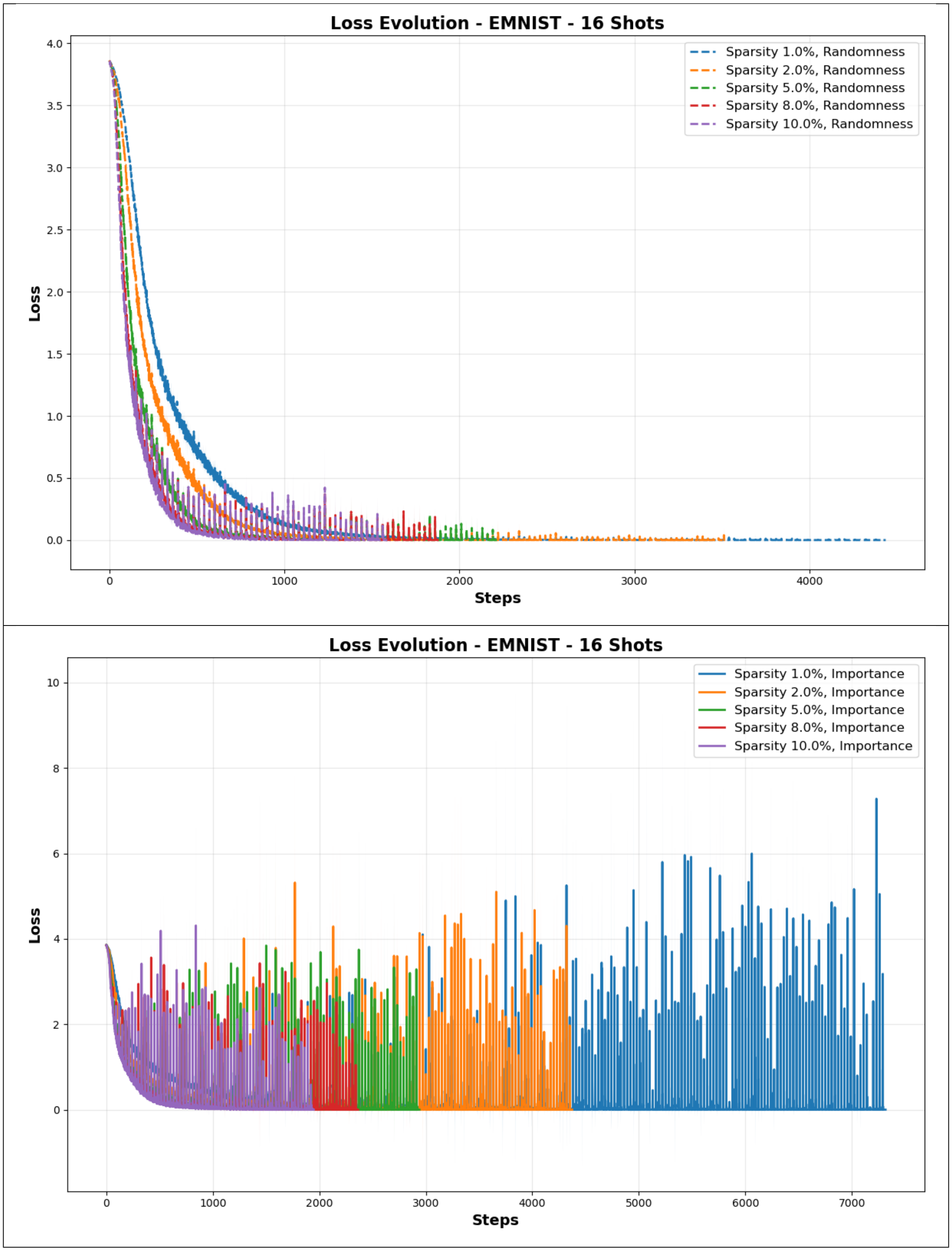


Figure 11. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Gradient Rank Evolution in Few-Shot Learning (4 shots) on MNIST Dataset.

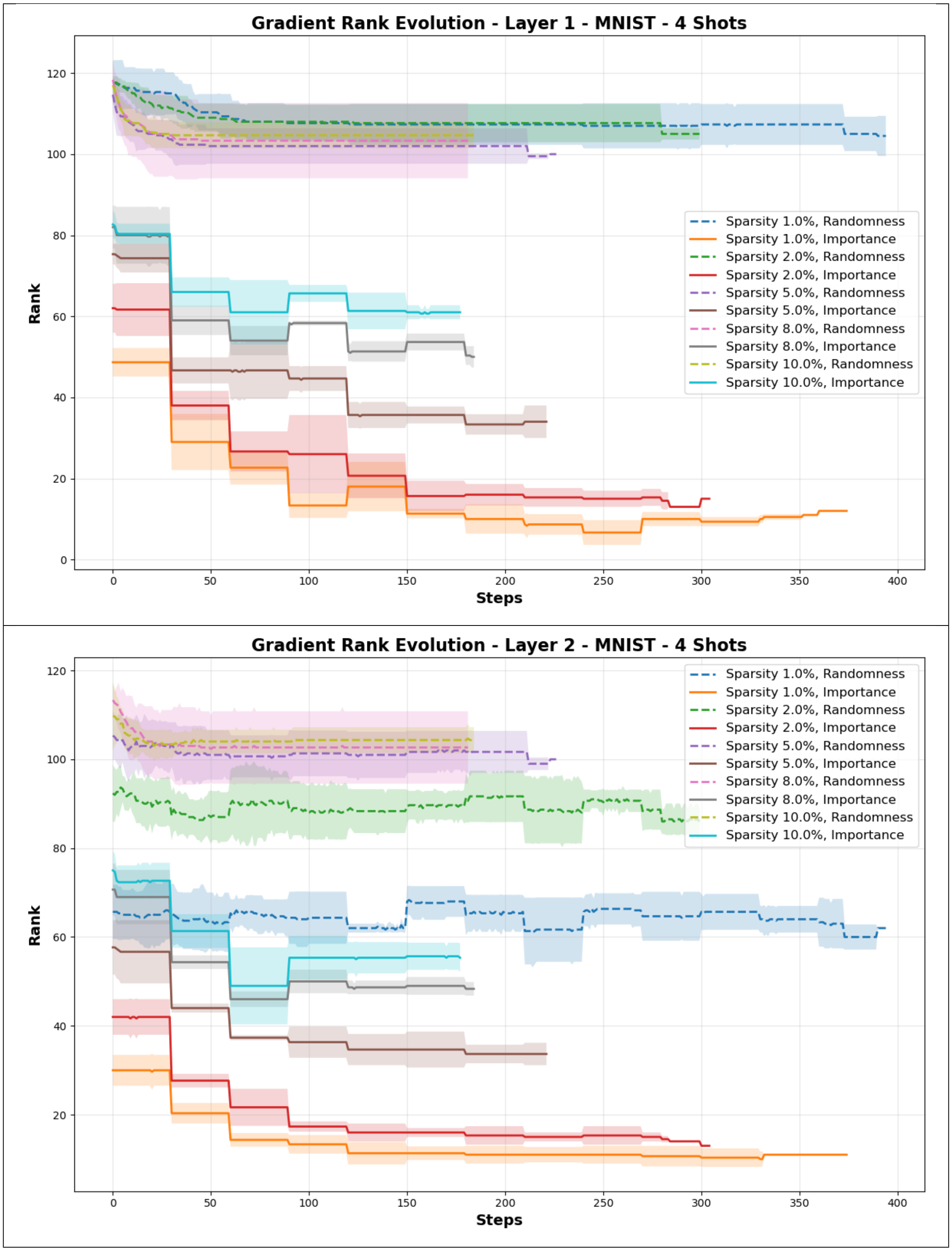


Figure 12. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Loss Evolution in Few-Shot Learning (4 shots) on MNIST Dataset.

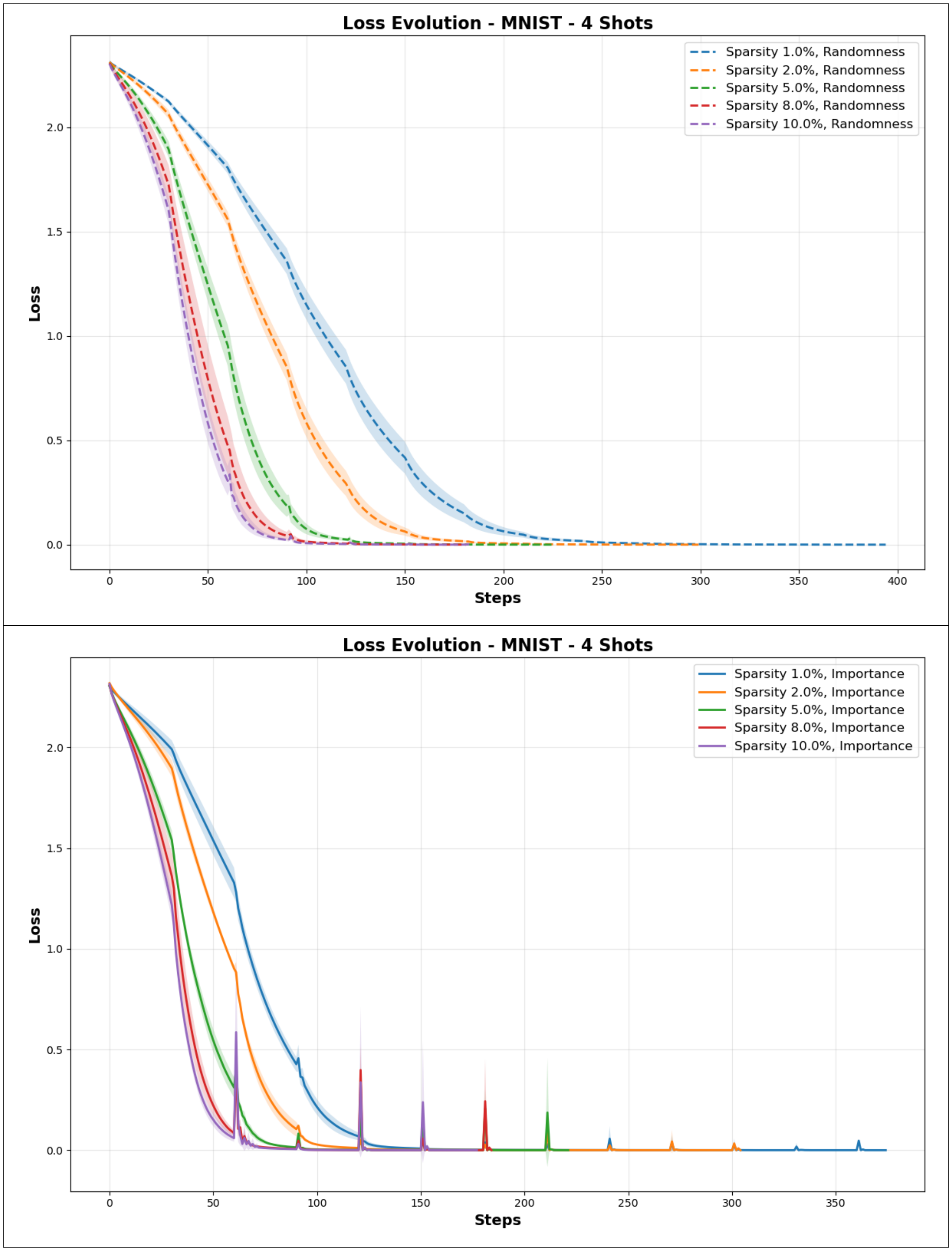
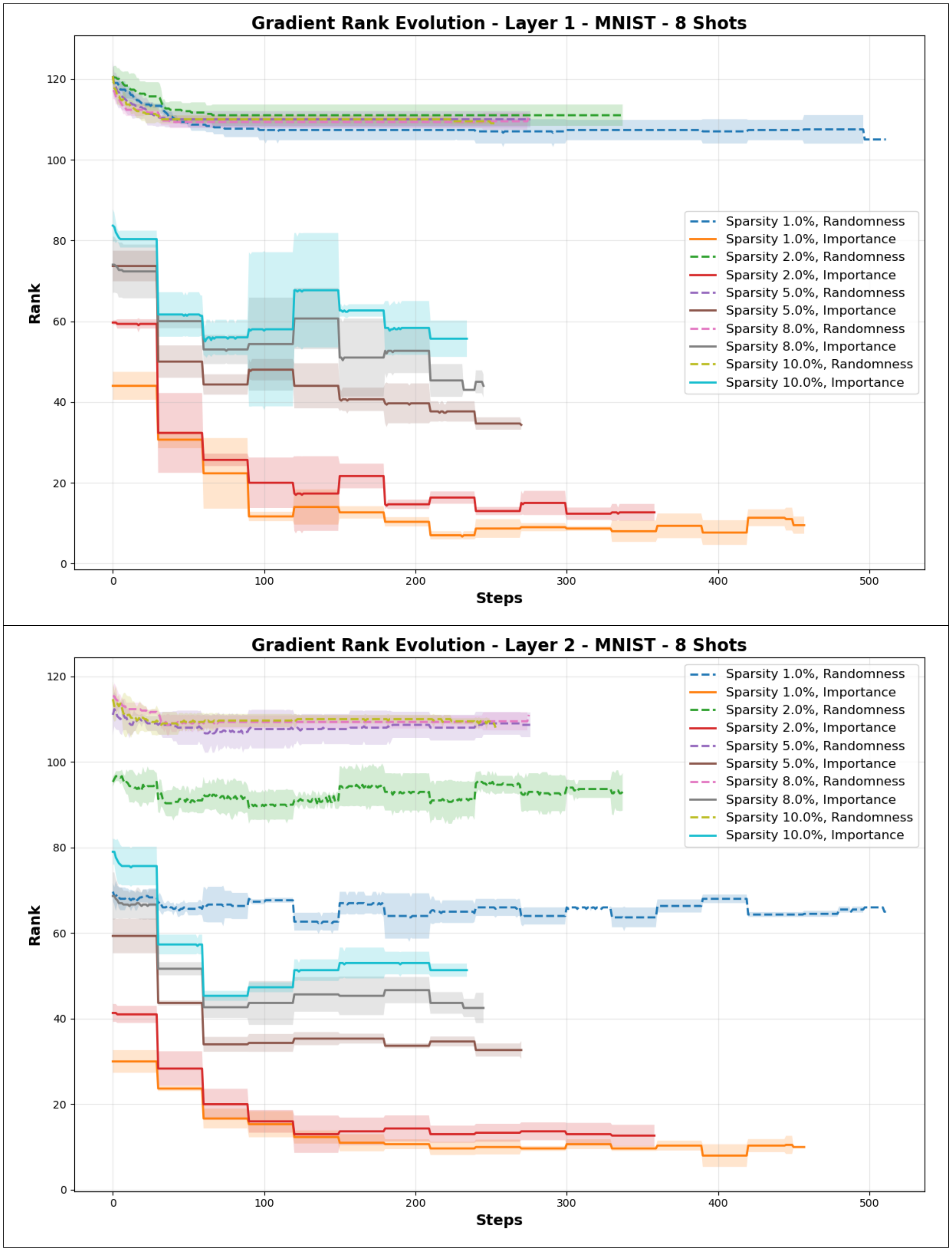


Figure 13. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Gradient Rank Evolution in Few-Shot Learning (8 shots) on MNIST Dataset.



Gradient Rank Evolution - Layer 1 - MNIST - 8 Shots

Gradient Rank Evolution - Layer 2 - MNIST - 8 Shots

Figure 14. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Loss Evolution in Few-Shot Learning (8 shots) on MNIST Dataset.

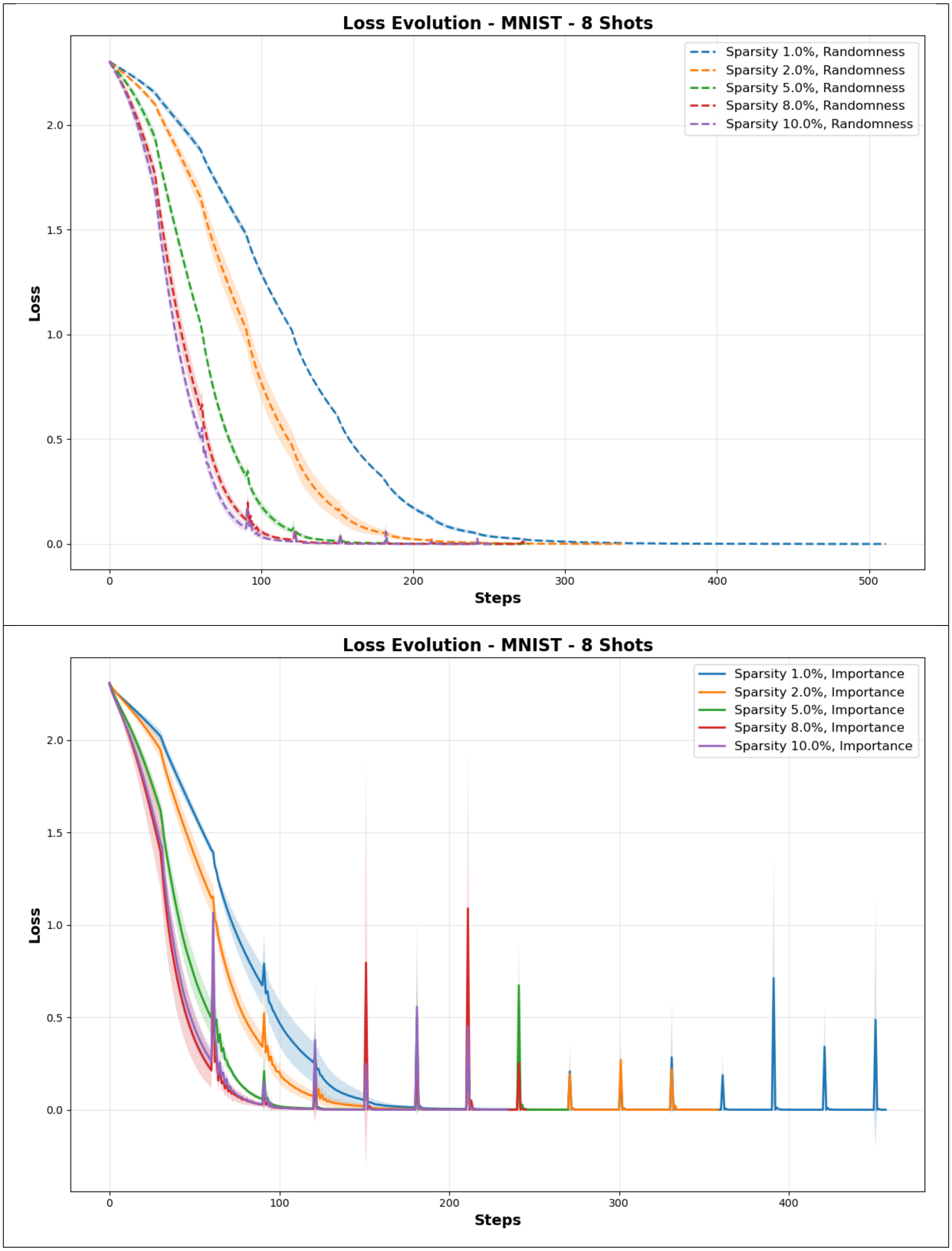


Figure 15. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Gradient Rank Evolution in Few-Shot Learning (16 shots) on MNIST Dataset.

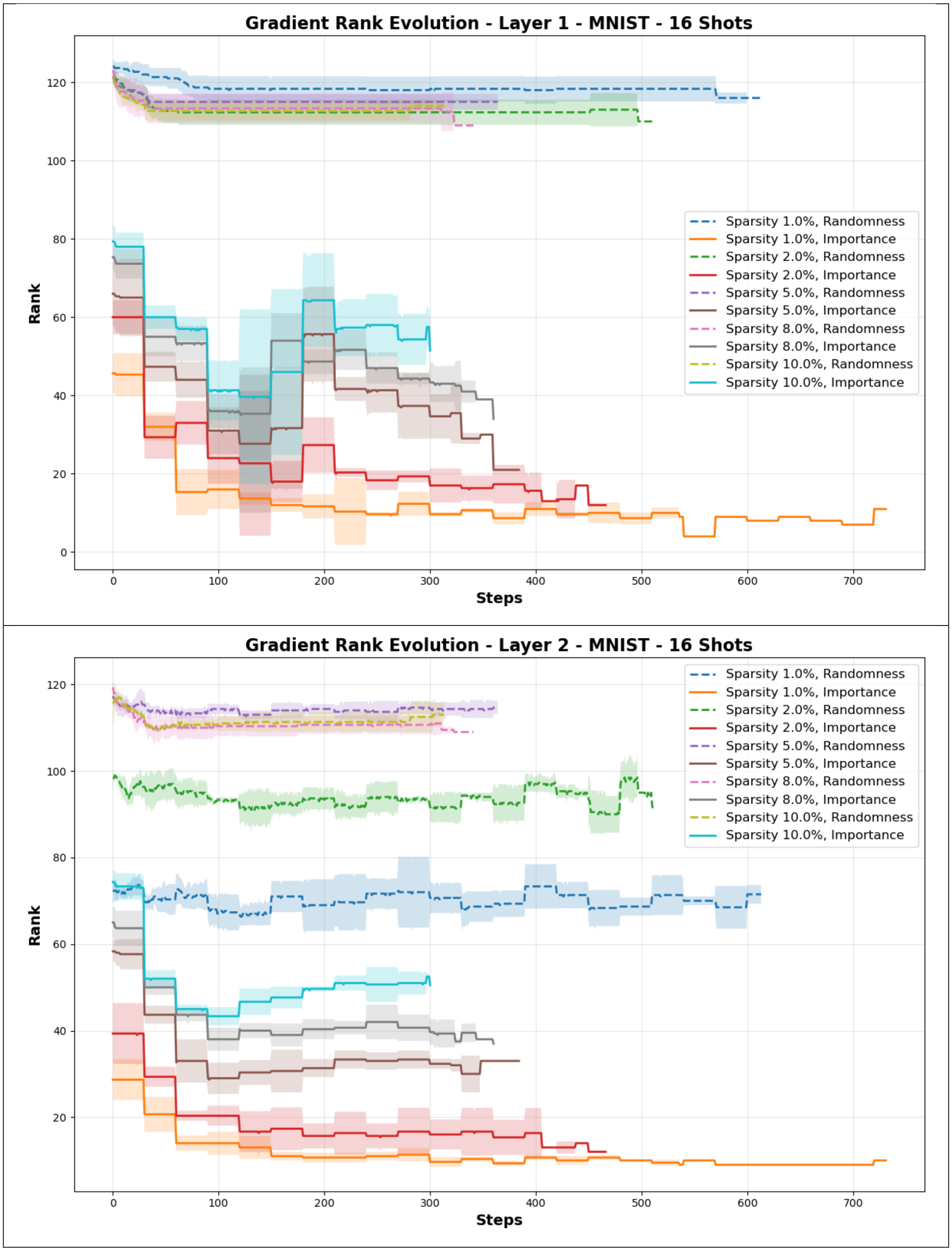
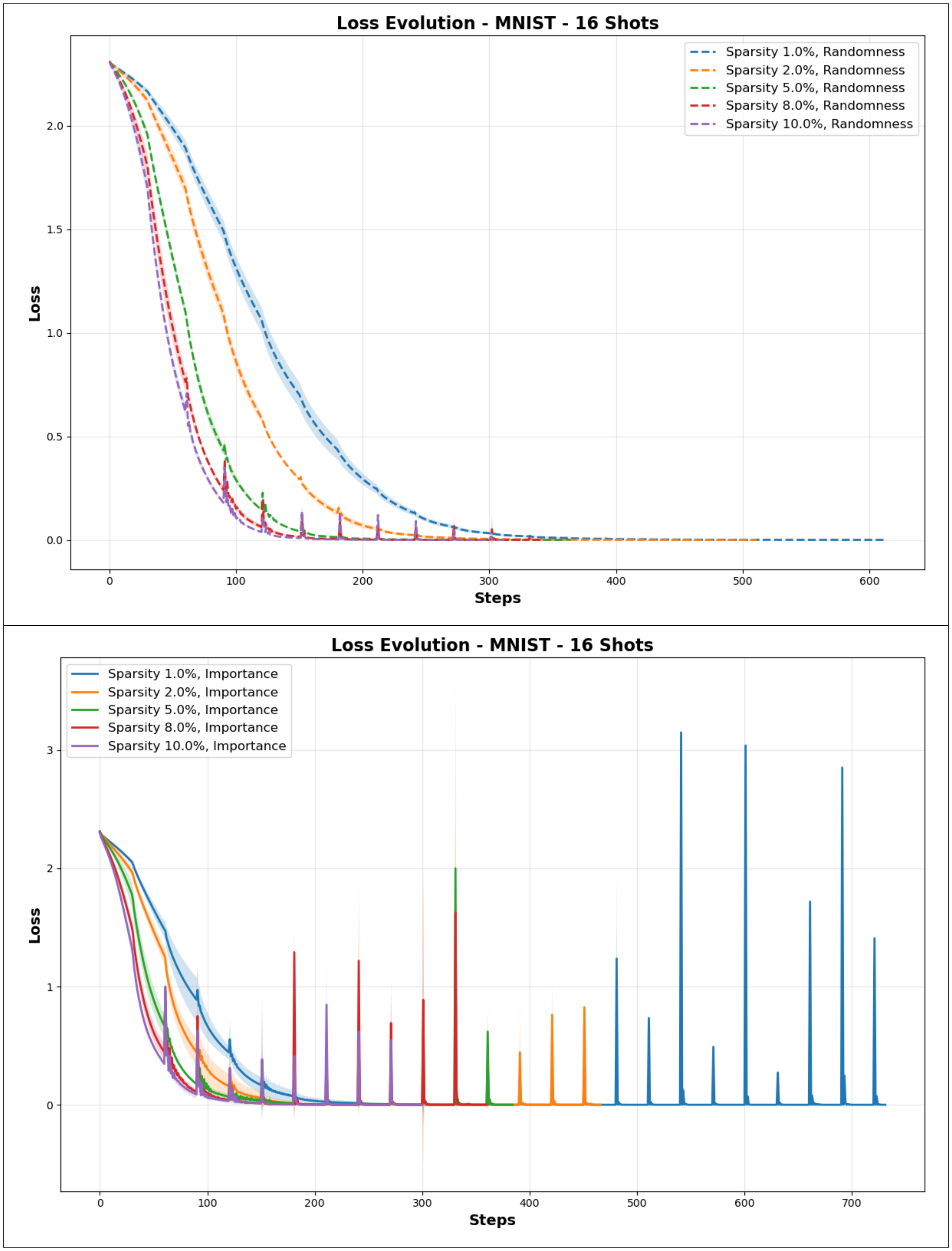


Figure 16. Comparison of Random and Importance Gradient Pruning in Sparse Optimization (SO) – Loss Evolution in Few-Shot Learning (16 shots) on MNIST Dataset.


Loss Evolution - MNIST - 16 Shots
Sparsity 1.0%, Randomness
Sparsity 2.0%, Randomness
Sparsity 5.0%, Randomness
Sparsity 8.0%, Randomness
Sparsity 10.0%, Randomness
Loss
Steps
Loss Evolution - MNIST - 16 Shots
Sparsity 1.0%, Importance
Sparsity 2.0%, Importance
Sparsity 5.0%, Importance
Sparsity 8.0%, Importance
Sparsity 10.0%, Importance
Loss
Steps