# Uncovering the Hidden Cost of Model Compression

## Supplementary Material

## 1. Background on Model Reprogramming

Mathematically, let $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ denote $n$ pairs of data samples $\{\mathbf{x}_i\}$ and their classification labels $\{\mathbf{y}_i\}$ for a target image classification task. $\mathbf{x}_i \in \mathbb{R}^{w \times h \times c}$ and $w$, $h$, $c$ are the image width, height, and number of color channels, respectively. $\mathbf{y}_i \in \{1, 2, \ldots, K\}$ and $K$ is the total number of image class labels. Let $f_\theta(\cdot)$ denote a pre-trained image classifier parametrized by $\theta$, which takes an image $\mathbf{x} \in \mathbb{R}^{w' \times h' \times c}$ as input and gives a prediction $f_\theta(\mathbf{x})$ of prediction probabilities over $K'$ classes, where $K \leq K'$, $w \leq w'$, and $h \leq h'$. In the standard VP training procedure, a masked perturbation, denoted by $\mathbf{M} \odot \delta$, is appended to a zero-padded version of $\{\mathbf{x}_i\}$ (denoted by $\{\mathbf{x'}_i\}$) in order to match the input dimension of the pre-trained model. The binary mask $\mathbf{M} \in \{1, 0\}^{w \times h \times c}$ denotes where to add the trainable perturbation to zero-padded images, and $\delta \in \mathbb{R}^{w \times h \times c}$ serves as a trainable universal perturbation. At the model output, a mapping function $h_k$ is assigned for every target class label $k \in \{1, 2, \ldots, K\}$ such that $h_k(f_\theta(\mathbf{x}' + \mathbf{M} \odot \delta))$ gives the prediction probability of the class $k$ for an image $\mathbf{x}$ in the target domain. Finally, VP trains the parameters associated with the input transformation (e.g. $\delta$) and/or the output mapping layers (e.g. if $\{h_k\}_{k=1}^K$ has trainable parameters) based on task-specific loss evaluated on $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$.

## 2. Experiment Details

In this section, we describe the hyperparameters used for all the experiments shown in **??** in the main text and the experiments in the appendix.

| **NS** | 1 | 2 | 5 | 10 | 20 | 50 | 100 | Full |
|---|---|---|---|---|---|---|---|---|
| **BS** | 8 | 8 | 16 | 32 | 32 | 64 | 128 | 128 |

Table 1. The different batch sizes (BS) used for each N-shots (NS) configuration for all experiments in **??** and Sec. 3

For all experiments discussed in **??** and 3, we train the model for 100 epochs using the Adam optimizer [**?** ]. We employ a multistep learning rate decay scheduler that reduces the learning rate by a factor of $\frac{1}{10}$ at the 50th and 72nd epochs, respectively, from its initial value of 0.01. The specific batch sizes used for training with each N-shots configuration are detailed in Table 1. To ensure reproducibility, we utilized checkpoints from [**?** ] for the RigL [**?** ] and AC/DC [**?** ] models, while the sparse checkpoints for ResNet-18, ResNet-34 and all the VGG [**?** ] variants were obtained from NeuralMagic SparseZoo. The dense checkpoints used were imported from the Torchvision library [**?** ].

In subsequent sections, we present additional experiments that go beyond the scope of the results outlined in the main text.

## 3. Additional LTH Experiments

**Lottery Ticket Hypothesis:** In this section, we study the performance of LTH solutions for ResNet-50 when transferred at various configurations of sparsity states at different data-budget settings. In the left subplot for each dataset, we report the test accuracy of the dense model and the sparsest model (at $\sim$12% sparsity) for transfer via the three VP methods, while each cell of the heatmap subplots on the right represents the mean difference across seeds between the dense model performance versus the sparse model transfer at a specified state of (sparsity, N-shot) pair. We only show the heatmaps for transfer via ILM-VP and FLM-VP in the main text and reflect on the RLM-VP heatmaps in the Supplement. Sec. 3 due to the generally poor performance of RLM-VP regardless of model sparsity or low data volumes compared to the other two VP methods. The upstream performance of the LTH solutions at different sparsity levels used in this study is shown in Figure 1. We show here the results on CIFAR-10, OxfordPets, DTD, and Caltech101 while the others are in supplementary Sec. 3.

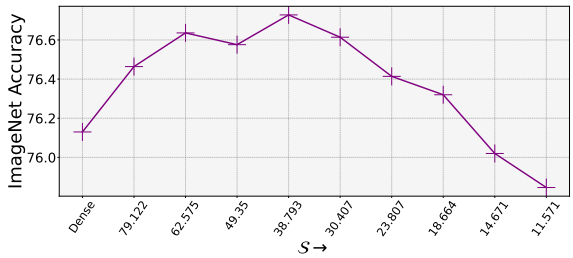



Figure 1. Top-1 accuracy of **LTH solutions** of a ResNet-50 pre-trained on the ImageNet-1k at different sparsity ($\mathcal{S}$) states.

Taking CIFAR-10 as an example, from the test-accuracy subplot in Fig. 2 (**top**) that for every N-shots budget, the dense model performed superior to that of the LT for transfer via ILM-VP and FLM-VP, and this trend holds true for all other datasets as well. Furthermore, we observe that ILM-VP outperforms FLM-VP in all N-shot settings. RLM-VP, which generally has a much lower performance compared to other VP methods, shows a slight deviation from this trend where we see that the sparse model tends to match the performance of the dense model, especially at transfer settings with a higher data budget.

Furthermore, we observe that the detrimental impact on

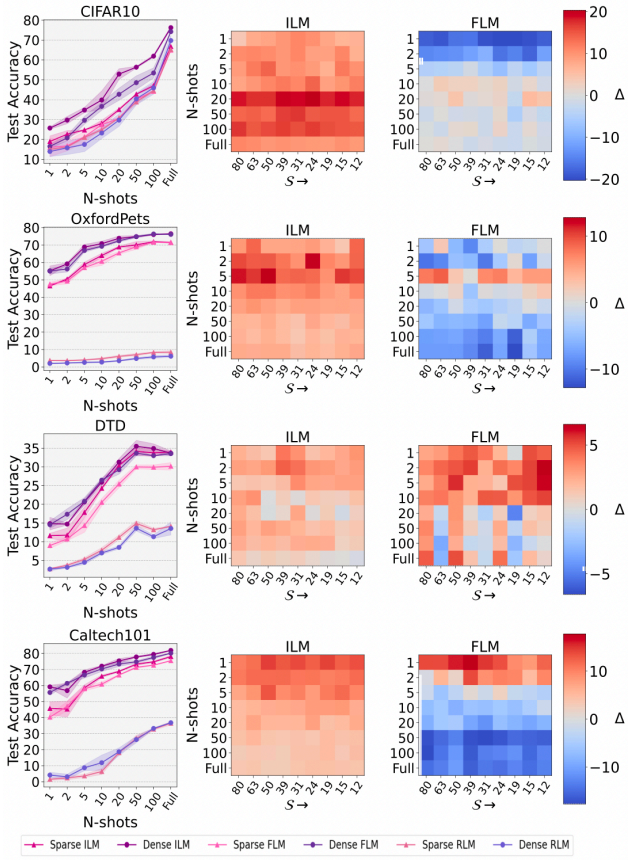



Figure 2. **Performance Gap of LT solutions on Transfer.** In each subfigure, from left to right, the first subplot represents the comparison between transfer via various VP methods for both the dense network and LT at $\approx 11\%$ sparsity on CIFAR-10 (**top**), OxfordPets, DTD and Caltech101 (**bottom**) at different target $N$-shot settings, while the two subplots on the right represent the $\Delta-$difference for ILM-VP and FLM-VP respectively. $\mathcal{S}$ represents increasing levels of %-sparsity levels of Dense (leftmost), 79.122, 62.575, 49.35, 38.793, 30.407, 23.807, 18.664, 14.671, and 11.571 (rightmost).

performance due to LTs was significantly more pronounced in the case of ILM-VP where, for example, in the case of 20-shot configurations, LTs in all sparsity states studied in this work had on average a 20% reduction in top-1 accuracy compared to their dense counterpart (see Figure 2 (**top**)). In the case of FLM-VP, the performance of LTs was actually better than that of the dense model for the few-shot settings as seen in the case of the one-shot and two-shot data budget settings; however, at higher data budget settings, the degradation of performance increases. The trend for ILM-VP transfer remains consistent across all four datasets, but there is a variation in that of FLM-VP based transfer. Specifically, for OxfordPets while the LTs overall seem to match or outperform their dense counterparts, for Caltech101 (see Figure 2 (**bottom**)) this only holds for the higher data budget settings. For DTD, on the other hand, barring a few data budget settings, the sparse model transfer seems to hurt the transfer performance overall.

We primarily base our conclusions on the trends for ILM-VP as it is the SOTA method, and thus in general, it is clear that the transfer of these LT solutions using VP-based methods does not keep their performance intact under low data volumes, although their upstream performance matches or outperforms their dense counterpart (see Figure 1).

In this section, we expand upon the findings of the lottery ticket hypothesis (LTH) discussed in Section Sec. 3. We present the results of transferring ResNet-50 LTH solutions using ILM-VP [**?** ] and FLM-VP [**?** ] to the four remaining downstream datasets: SVHN [**?** ], GTSRB [**?** ], Flowers102 [**?** ], and EuroSAT [**?** ]. Subsequently, we evaluate the transfer performance of RLM-VP on all eight datasets.

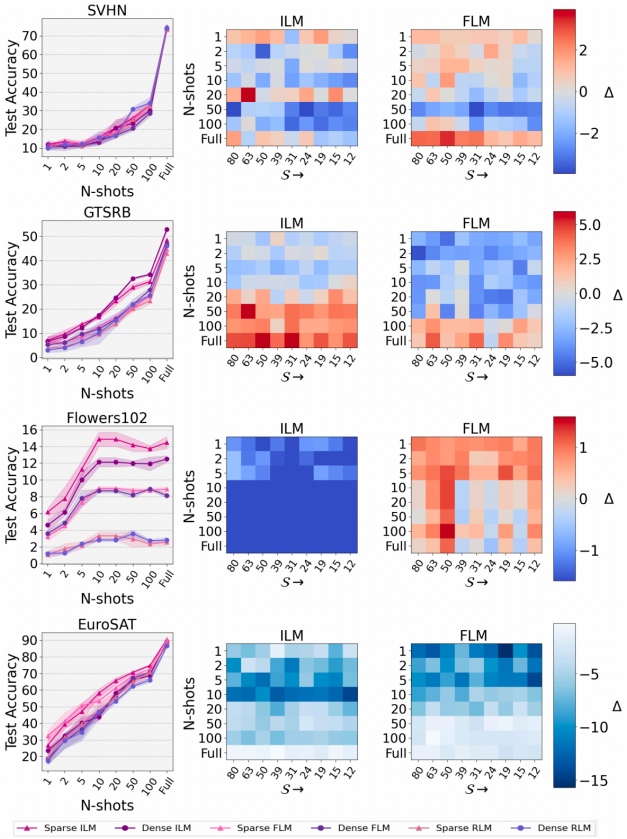



Figure 3. **Performance Gap of LT solutions on Transfer.** In each subfigure, from left to right, the first subplot represents the comparison between transfer via various VP methods for both the dense network and LT at $\approx 11\%$ sparsity on SVHN (**top**), GTSRB, Flowers102, and EuroSAT (**bottom**) at different target $N$-shot settings, while the two subplots on the right represent the $\Delta-$difference for ILM-VP and FLM-VP respectively. $\mathcal{S}$ represents increasing levels of %-sparsity levels of Dense (leftmost), 79.122, 62.575, 49.35, 38.793, 30.407, 23.807, 18.664, 14.671, and 11.571 (rightmost).

From the heat maps depicted in Figure 3, a notable contrast emerges when comparing the performance on SVHN, GTSRB, and Flowers102 with the four datasets discussed in Sec. 3 in the main text. On these three datasets, the discrepancy between the sparse and dense models is relatively small, generally within the range of approximately 5%.

For SVHN, it is observed that sparse model transfer yields a performance dip compared to its dense counterpart in sporadic instances of low- and high-data volumes, particularly noticeable in the case of FLM-VP. However, in other data volume scenarios, the sparse model either matches the dense model or slightly outperforms it, regardless of the visual prompting (VP) method employed.

In the case of GTSRB, under higher data volumes, transferring the sparse model results in performance degradation compared to the dense models for both ILM-VP and FLM-VP. Conversely, for Flowers102, while the transfer of sparse models through FLM-VP incurs performance loss across nearly all sparsity and data volume settings, ILM-VP exhibits a reversed trend. Here, sparse models outperform the dense counterpart by approximately 1% in high data volume settings and by around 0-1% in lower data volume settings.

For EuroSAT, it is observed that, for both FLM-VP and ILM-VP, the sparse model outperforms the dense counterpart, particularly in low-data-volume scenarios.

The accompanying test accuracy plot (leftmost) in Figure 3 reveals that, except for Flowers102, the performance of sparse and dense model transfer under all three visual prompting methods is closely aligned, with ILM-VP often exhibiting a slight edge over the other two methods.

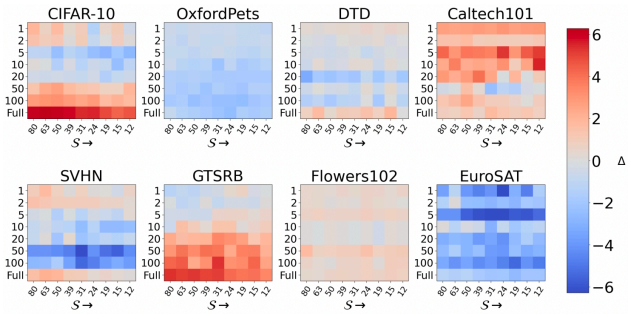



Figure 4. **Performance of LT solutions on Transfer via RLM-VP.** $\Delta-$difference for RLM-VP transfer of LTH solutions. $\mathcal{S}$ represents increasing levels of %-sparsity levels of Dense (leftmost), 79.122, 62.575, 49.35, 38.793, 30.407, 23.807, 18.664, 14.671, and 11.571 (rightmost).

Subsequently, we examine the transfer performance using RLM-VP in Figure 4 across the eight datasets. It is essential to highlight, based on the accuracy plots encompassing various model architectures, data volumes, and sparsity settings thus far, that RLM-VP consistently emerges as the least effective visual prompting (VP) method. It is frequently outperformed by a considerable margin compared to both FLM-VP and ILM-VP.

Across almost all datasets, with the exceptions of EuroSAT and GTSRB, the performance trends indicate a close similarity between the lottery ticket hypothesis (LTH) solutions and dense models, with sparse model transfer generally resulting in performance deterioration. In EuroSAT, similar to transfers using ILM-VP and FLM-VP, even under RLM-VP, the sparse model outperforms the dense model by approximately 2-5%.

In summary, considering the LTH solution trends presented in both the main text and this section, it becomes evident that, for the state-of-the-art visual prompting (VP) method ILM-VP and, for the most part, across various data and sparsity configurations of FLM-VP, sparse model transfer typically leads to performance degradation compared to their dense counterparts. For RLM-VP, a mixed trend is observed, though the interpretation is challenging due to the consistently poor transfer performance under this VP method.

## 4. Additional CLIP Experiments

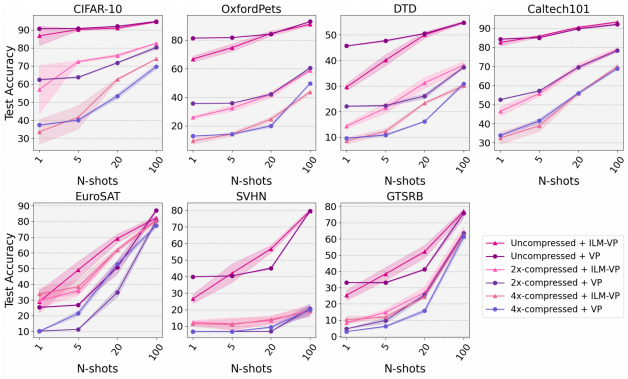



Figure 5. Transfer performance of uncompressed and compressed variants of CLIP across different N-shots configurations on a range of downstream datasets.

In this section, we extend our analysis beyond the results presented in **??** of the main text, where the reported results pertain only to the full data volume setting. For consistency with the settings of the other experiments detailed in this manuscript, we apply the same N-shot variability set-up to three different variants of CLIP [**?** ] - uncompressed, 2x compressed and 4x compressed - obtained through the UPop [**?** ] compression method. Using the identical hyperparameter configuration outlined in Section **??**, the results presented in Figure 5 reaffirm the trends observed in the full data volume setting from the main text. Across all N-shot settings within the seven downstream datasets used, a consistent decline in performance is observed when transitioning from the uncompressed CLIP variant to the 2x and 4x compressed variants.

With the exception of EuroSAT, there is a statistically significant performance gap between the uncompressed and compressed variants across all datasets. Across all settings, the performance follows the pattern: uncompressed variant $\gg$ 2x compressed variant $>$ 4x compressed variant. For example, in the OxfordPets dataset, we note a decrease in accuracy exceeding 40% and 65% when comparing the uncompressed variant to the 2x compressed and 4x compressed variants, respectively, at N-shot values of 1 and 5.

#### 4.0.1 What Leads to the Hidden Cost in VP?

The aforementioned results on different model compression methods and sparse vision models unveil the existence of a common weakness in the severely degraded performance of VP. We hypothesize that the observed degradation is a hidden cost of model compression that accidentally weakens the label-mapping capability of the original model in VP. To verify this hypothesis, we conduct the following experiments and analyses to track the changes of VP in label mapping and training dynamics.

To explore the differences between visual prompting with compressed models and their dense, full-precision counterparts, extending beyond accuracy, we begin by examining the label mapping process under ILM-VP for a ResNet-50 LT ($\mathcal{S} = 14.671$) in a few-shot setting (N=5) on the OxfordPets [**?** ] and DTD [**?** ] datasets. As depicted in Figure **??**, our analysis reveals that the dense model maps the 'Bombay' class from the OxfordPets dataset to the 'Schipperke' class from the ImageNet [**?** ] dataset, establishing a semantically closer mapping. On the contrary, the sparse LT model maps the same class to the 'Carton' class from the ImageNet dataset, a less semantically related mapping.

Furthermore, for the target dataset DTD, the dense model maps the 'Zig-Zagged' class to the 'Chiffonier' class of the dataset ImageNet. While the object categories do not directly correspond, it can be argued that zig-zag textures are more prevalent on the furniture texture frames of chiffoniers, as evident from the second example of the 'Zig-Zagged' class compared to the third example of the 'Chiffonier' class. This highlights a critical drawback of sparse models, indicating that they suffer from inferior label mapping, which ultimately hinders downstream performance.

A comparable pattern emerges in the context of model quantization. For example, in the case of the target class "Sphynx" (a cat breed) from the OxfordPets dataset, the 2-bit quantized version of DeiT incorrectly assigns it to the unrelated label "tub," whereas the full-precision 32-bit variant of DeiT accurately maps it to the semantically related class "Mexican Hairless" (a dog breed) from the source ImageNet dataset.

To accurately characterize and distinguish the class-wise performance of visual prompted sparse and dense models,

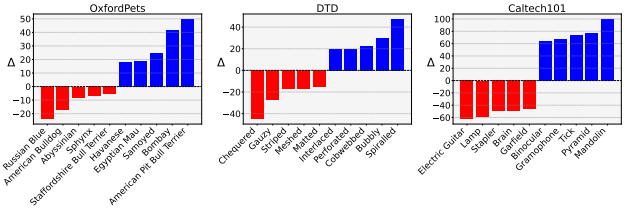
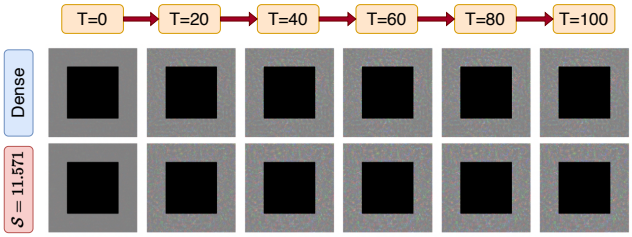
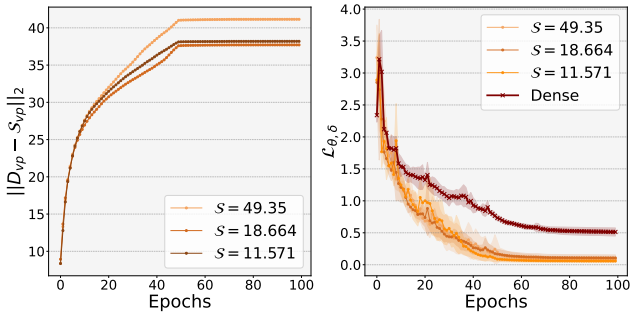



Figure 6. **Class-wise Performance Analysis.** Classes with Top-5 drop and gains in accuracy of transfer via ILM-VP on the three datasets of OxfordPets, DTD, and Caltech101. $\Delta$ represents the mean difference in top-1 accuracy of the dense model and the sparse model for each of the classes.




(a) **Evolution of VP.** Visual prompt pattern versus number of training epoch for a ResNet-50 dense model and sparse LTH solution ($\mathcal{S} = 11.571\%$). (Best viewed when zoomed in)




(b) From left to right: (i) Change in $L_2$-norm of the difference between the learned visual prompts of a dense ResNet-50 model and its sparse LTH solutions variants at varying sparsity. (ii) **Convergence Analysis**: Training loss trajectories of the dense and sparse LTH solutions.

Figure 7. Training dynamics of Visual Prompting for ResNet-50 dense and sparse LTH variants on the Caltech101 dataset under few-shot settings (N=5).

we conduct an analysis across three datasets: (a) OxfordPets [**?** ], DTD [**?** ], and Caltech101 [**?** ]. Specifically, we identify the top five classes in which the sparse model outperforms the dense counterpart and vice versa. Using the sparse models featured in Figure **??**, we analyze visual prompted dense ResNet-50 and sparse LT ResNet-50 ($\mathcal{S} = 14.671$) trained in a few-shot setting (N=5). As illustrated in Figure 6, our results indicate that while certain classes exhibit improved performance with the sparse model, the magnitude of this improvement is often less pronounced compared to the performance advantage of the dense model. For example, in the

case of the Caltech101 dataset, the dense model achieves a 100% increase in accuracy for the 'Mandolin' class when compared to the sparse LT model. Although these findings offer preliminary insights into the class-wise performance dynamics between sparse and dense models, a more comprehensive understanding of these dynamics is deferred to future investigations.

Finally, we examine the evolutionary trajectories and training dynamics of the visual prompts learned by sparse models compared to their dense counterparts. To conduct this analysis, we consider both ResNet-50 dense and sparse LTH solutions ($\mathcal{S} = 11.571, 18.664, 49.35$) trained on the Caltech101 dataset in a few-shot setting (N=5). The insights gained from this examination are illustrated in Figure 7.

As training progresses, we observe a gradual increase in the $L_2$-norm of the difference between the visual prompt learned by the dense model and each of the sparse model variants. This analysis demonstrates how the visual prompt of the sparse models with more compute diverges from the more optimal visual prompt learned by the dense model.

## 5. Additional Experiments

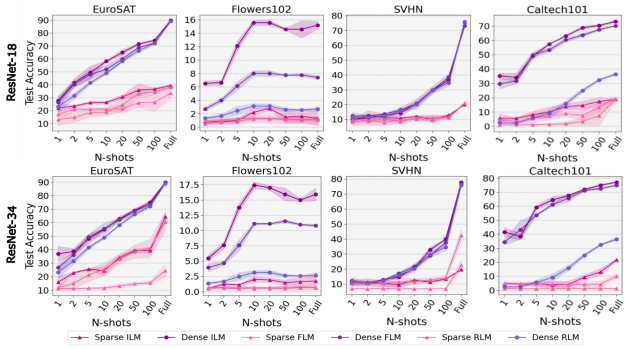



Figure 8. **GMP-pruned ResNet-18/34.** Transfer performance measured by test accuracy of pruned ResNet-18/34 model on a variety of downstream datasets and varying levels of data budgets.

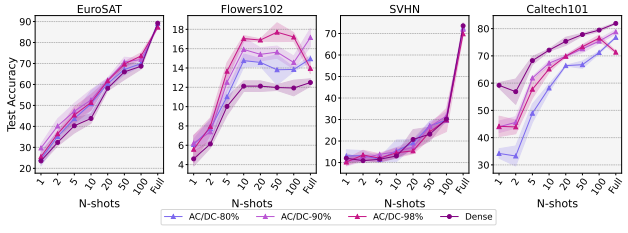



Figure 9. **AC/DC-pruned ResNet-50.** Transfer performance measured by test accuracy of pruned ResNet-50 model on a variety of downstream datasets and varying levels of data budgets.

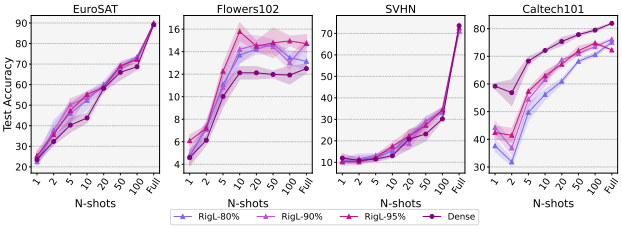



Figure 10. **RigL-pruned ResNet-50.** Transfer performance measured by test accuracy of pruned ResNet-50 model on a variety of downstream datasets and varying levels of data budgets.

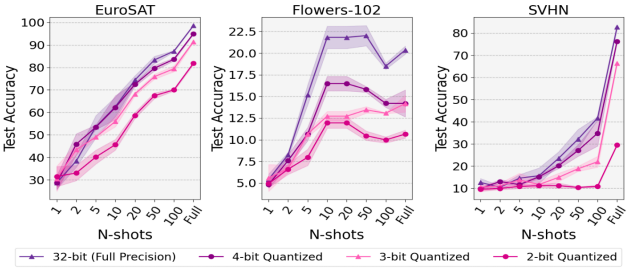



Figure 11. **VVTQuantized DeiT-T.** Transfer performance measured by test accuracy of quantized DeiT-T models on a variety of downstream datasets and varying levels of data budgets.

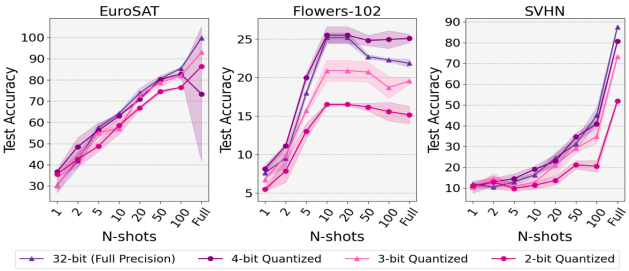



Figure 12. **VVTQuantized Swin-T.** Transfer performance measured by test accuracy of quantized Swin-T models on a variety of downstream datasets and varying levels of data budgets.