A. Hyperparamter Details for Main Results

Hyperparameter	Range
optimizer	ADAMW
learning-rate, (lr)	$\{0.005, 0.001, 0.0005, 0.0001\}$
weight-decay, (wd)	$\{0.0001, 0.001\}$
epochs	100
warmup epochs	10

Table 3. **Hyperparameter Range**

The hyperparameters and their ranges used in our main experiments are tabulated in Table 3. In Table 4, we report the original results of Table 1 with optimal number of prompts per task.When comparing to other prompting techniques like VPT, we use the official implementation[1]. We observe that in many tasks (like SVHN, Patch-Cam., Clevr/distance, Kitti/dist., dsprites/orient.), EXPRES outperforms VPT-deep with significantly fewer prompts.

B. Effect of Hyperparameters

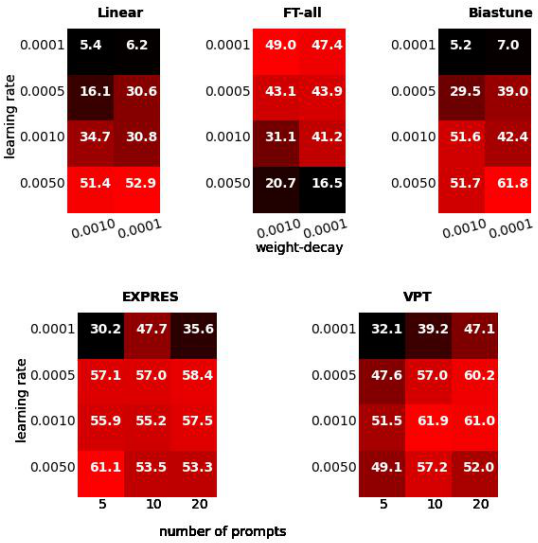

Figure 6. **Effect of hyperparameters for Five-Shot Semantic Segmentation on PASCAL -5^0**

We study the effect of hyperparameters on various baselines and EXPRES used for five-shot segmentation. To create the validation set for the i^{th} fold, we sample 100 random tasks from the corresponding train-split that consists of fold exclusive categories, $\{5^j | j \neq i\}$ and evaluate the average accuracy over these tasks. For each baseline method, we test the following learning rates: $\{0.0001, 0.0005, 0.001, 0.005\}$. For prompt based methods, we fix the weight decay to 0.0001 and test following number of prompts: $\{5, 10, 20\}$. For rest of the methods (non-prompt-based), we vary the weight decay in the set, $\{0.001, 0.0001\}$. In Figure 6, ADAMW is chosen as the optimizer and fold-5^0 as the fold for analysis. For other hyperparameters like number of epochs and warmup epochs, fixed values of 100 and 10 respectively worked well. We use the above hyperparameters validation to pick the optimal values for final evaluation. In Figure 7, we visualize the predictions for EXPRES and compare it to various baselines. EXPRES tends to produce more complete segmentation masks than others across different folds.

[1] https://github.com/KMnP/vpt

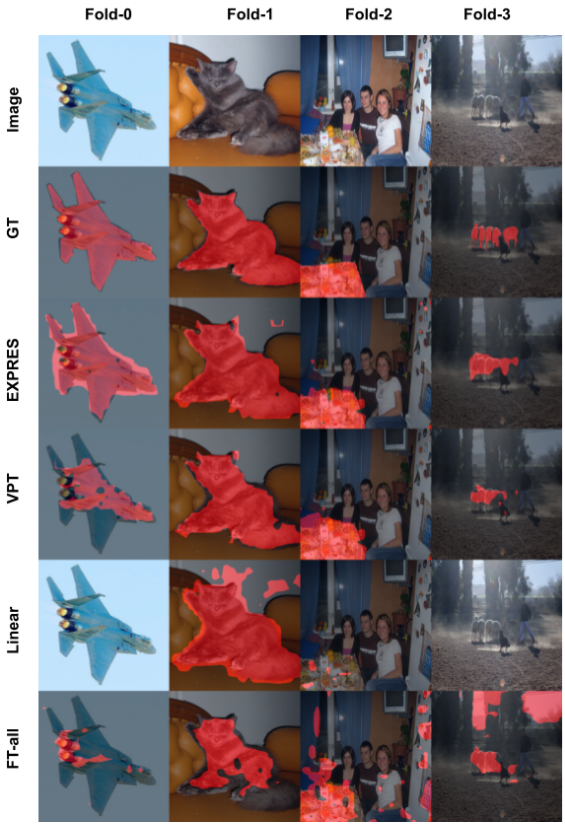

Figure 7. Five-Shot Predictions on PASCAL -5^i

C. Additional Ablations on VTAB-1k

We demonstrate the importance of **prompt propagation** and **residual prompts** on additional tasks (*DTD* and *Clevr-Count*) from the VTAB-1k benchmark. In all ablations, we use optimal learning rates and weight decays with $M = 15$ for *DTD* task and $M = 10$ for *Clevr-Count* task. In Figure 8, we repeat the ablation of propagating shallow prompts without residual prompting for additional datasets. We observe similar trends as in Figure 3- downstream perfor-

	Natural								Specialized					Structured								
	CIFAR-100	**Caltech101**	**DTD**	**Flowers102**	**Pets**	**SVHN**	**Sun397**	**Mean**	**Patch Camelyon**	**EuroSAT**	**Resisc45**	**Retinopathy**	**Mean**	**Clevr/count**	**Clevr/distance**	**DMLab**	**KITTI/distance**	**dSprites/location**	**dSprites/orientation**	**SmallNORB/azimuth**	**SmallNORB/elevation**	**Mean**
VPT-shallow	77.7	86.9	62.6	97.5	87.3	74.5	51.2	76.81	78.2	92.0	75.6	72.9	79.66	50.5	58.6	40.5	67.1	68.7	36.1	20.2	34.1	46.98
num. prompts (M)	100	5	1	200	50	200	1	79.4	5	50	50	10	28.7	100	200	100	100	100	100	200	200	137.5
VPT-deep	**78.8**	**90.8**	65.8	98.0	88.3	78.1	49.6	78.48	81.8	96.1	83.4	68.4	82.43	**68.5**	60.0	**46.5**	72.8	73.6	47.9	**32.9**	**37.8**	54.98
num. prompts (M)	10	10	10	1	1	50	5	12.4	100	100	10	1	52.8	50	200	100	50	10	50	200	200	107.5
EXPRES (*ours*)	78.0	89.6	**68.8**	**98.7**	**88.9**	81.9	**51.9**	**79.7**	**84.8**	**96.2**	80.9	**74.2**	**84.0**	66.5	**60.4**	**46.5**	**77.6**	**78.0**	**49.5**	26.1	35.3	**55.0**
num. prompts (M)	30	10	15	10	5	10	10	12.9	30	10	10	10	15.0	100	10	10	10	30	30	10	30	28.75

Table 4. **Extended VTAB-1k results:** Comparing EXPRES with VPT-shallow and VPT-deep (VPT) with optimal number of prompts per VTAB-1k task. The highest accuracies are highlighted per task.

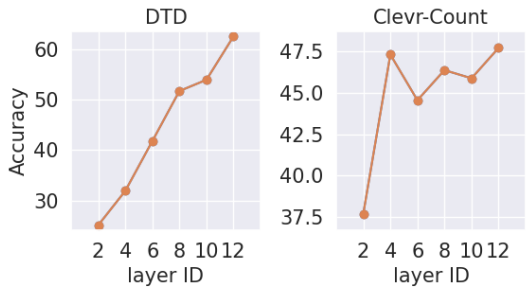

Figure 8. **Prompt propagation (additional ablations):** Effect of propagating prompts with modulation upto a layer, $l = \{2, \ldots, 12\}$ of the ViT-B/16 encoder with total 12 layers. The datasets are sampled from the VTAB-1k benchmark.

mance depends directly on the extent of prompt propagation with modulation *i.e.*, interaction with other tokens via self-attention. We also observe that for *Clevr-Count*, the performance quickly rises upto layer 4, then plateaus. Thus, the exact performance trend with increasing propagation through the layers varies slightly with the downstream task. In Figure 9, we repeat the ablation that delineates the importance of each type of residual prompt for additional datasets. We observe that trends are similar to Figure 4. Within each block (*Att* and *MLP*), layer norm residual prompts yield notable improvements in most tasks. Within MSA block (*Att*), the *QKV* prompts are dominant over *Proj* prompts for *natural* tasks (DTD, svhn) of VTAB-1k while the trend reverses for *structured* tasks (Clevr-Dist, Clevr-Count). Most importantly, when comparing blockwise performance, prompting the MSA block (*Att*) consistently outperforms prompting the MLP block (*MLP*), reinforcing the conclusions in §4.3.

D. Effect of Prompt Initiation Layer

One of the key design decisions of EXPRES prompting is where to insert the prompts. In Fig. 10 we show that initiating EXPRES prompting at early layers is generally a good strategy. Combined with the observations in 3, we recommend using prompts at every layer to ensure best performance.

E. Effect of Architectural Choice

	Linear	MLP-3	Partial-1	Biastune	VPT	VPT-shallow	EXPRES
VTAB-*Specialized*	80.8	75.2	81.7	80.1	84.5	82.5	**84.6**

Table 5. **Comparing various methods for adapting Swin-Base model pretrained on ImageNet-21k**

EXPRES was designed as a generic prompting technique for Transformer architectures. To demonstrate it's generality, we compare our method with other adaptation techniques with Swin [51] transformers as the backbone. In particular, we incorporate EXPRES prompts in Swin-Base architecture, where fixed number (10) of shallow prompts are propagated through all four stages without any modification during patch merging with residual prompts being added at each layer of the SwinTF-blocks. Evaluations on a subset of the VTAB-1k dataset in Table 5 shows that overall our method outperforms various baselines and state-of-the-art methods even when applied to a different variant of transformer architecure.

F. FGVC results

In Table 6, we compare the performance of our approach to adaptation baselines and other prompting techniques on the FGVC benchmark. On four datasets our method outperforms most baselines and performs competitively with other prompting techniques.

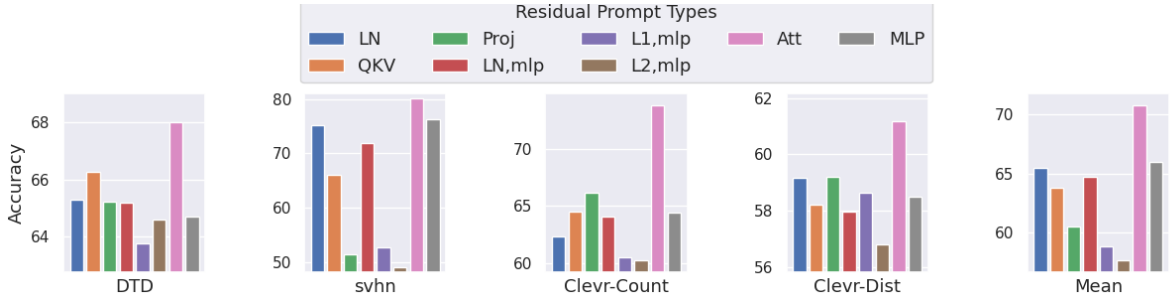

Figure 9. **Residual Prompt Types (additional ablations):** Evaluating the importance of different type of residual prompts on VTAB-1k datasets.

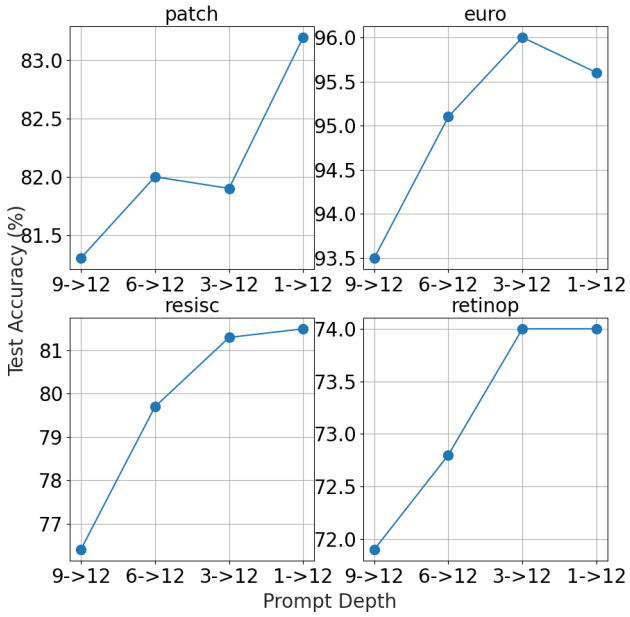

Figure 10. **Starting EXPRES prompting at specific layers:** $l_1 \rightarrow l_2$ means prompting starts at l_1 (closer to input) and ends at l_2. number of prompts is fixed at 10.

	Cub	Flower	Dogs	Cars
FULL	87.3	98.8	89.4	**84.5**
LINEAR	85.3	97.9	86.2	51.3
PARTIAL-1	85.6	98.2	85.5	66.2
MLP-3	85.1	97.9	84.9	53.8
SIDETUNE	84.7	96.9	85.8	48.6
BIAS	88.4	98.8	**91.2**	79.4
VPT-shallow	86.7	98.4	90.7	68.7
VPT	**88.5**	**99.0**	90.2	83.6
EXPRES (*ours*)	88.3	**99.0**	90.0	80.5

Table 6. **FGVC benchmark:** Per task adaptation results with ViT-B/16 model.

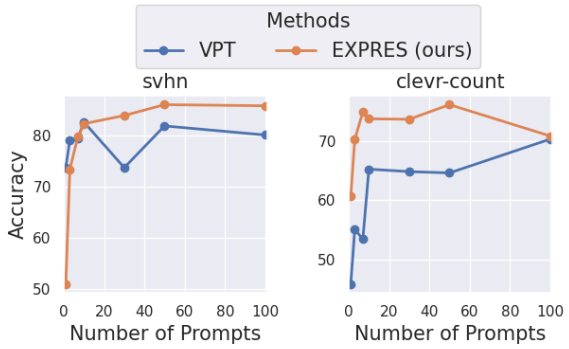

Figure 11. **Additional Computational Efficiency of EXPRES** Comparing Accuracy vs Number of Prompts for VPT and EXPRES

G. Additional Discussion on Performance-Cost Tradeoff

In Figure 11, we compare the computational efficiency of our EXPRES with VPT on additional VTAB-1k datasets. We fix the learning rate and weight decay for EXPRES at $lr = 0.01, wd = 0.0001$ and for VPT at the optimal values from [32]. Rest all hyperparameters like batch size, epochs and warmup epochs are left unchanged. We observe trends similar to Figure 5, where for a given accuracy specification, the difference between number of prompts required for EXPRES and VPT can be upto an order (*e.g.*, clevr-count in Figure 11).

In Table 7, we compare the computational as well as memory cost of various adaptation techniques. The computational cost is reported in terms of GMACs and memory cost in terms of tuned parameters relative to full model parameters. We observe that with 100 prompts, our EXPRES requires memory (4.7M params.) that is comparable to VPT (1M params.) and orders of magnitude less than finetuning ($\sim$ 86M params.). While the computational cost of our method (26.9 GMACs) is slightly more than Linear (17.5 GMACs) and VPT (19.9 GMACs) when using large num-

	Tuned Params (%)	GMACs
FT-all	100.0	17.47
Linear	0.090	17.47
VPT-shallow (M=1)	0.091	17.56
VPT-shallow (M=100)	0.179	26.87
VPT (M=1)	0.100	17.50
VPT (M=100)	1.166	19.96
EXPRES (M=1)	0.144	17.56
EXPRES (M=100)	5.560	26.87

Table 7. **Memory and computational cost analysis using a ViT-B/16 pre-trained on supervised ImageNet-21k**. We consider input resolution of $224 \times 224 \times 3$ and a 100-way classification task. Tuned parameters are reported as a percentage of the parameters in the backbone model (including classifier head). Here, M is number of prompts per layer.

ber (100) of prompts, our method greatly outperforms Linear (Table 1) and achieves better optimal performance than VPT with far fewer prompts (Figure 5 and Figure 11), providing good performance-cost tradeoffs.

H. Ablations for Semantic Segmentation

	Q	K	MLP
Accuracies	44.39	**51.61**	38.25

Table 8. **Ablation:** Effect of various representations for five-shot semantic segmentation on PASCAL $-\mathbf{5^0}$

We evaluate various ways of the extracting dense representations from transformer backbone for five-shot semantic segmentation. In Table 8, we compare last-layer keys (K), queries (Q) and MLP-block(MLP) outputs. We use a learning rate of 0.005, weight decay of 0.0001, 5 prompts and average the accuracies over 100 tasks randomly sampled from the train-split of PASCAL $-\mathbf{5^0}$. We observe that keys (K) are the most effective representations for semantic segmentation, so we use them in all our segmentation experiments.

I. Interpretability of Learnt Prompts

In Figure 12, we provide visualizations to demonstrate that residual prompts learn semantic information and facilitate fine-grained layerwise modulation of the attention. Here, an arbitrarily chosen but fixed prompt location is used for visualization purposes. We observe that residual prompts learn spatially fine-grained details that are diverse across layers and that removing them reduces the diversity of the attention maps across layers, confirming our hypothesis.

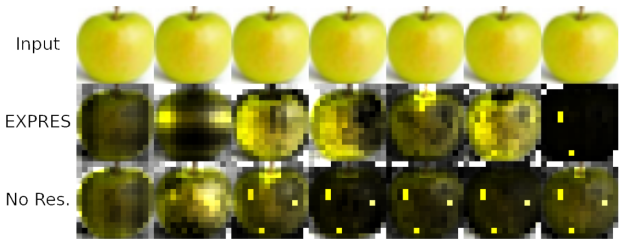

Figure 12. **EXPRES prompt attention at different layers:** We display input image (top-row), the attention maps of a trained EXPRES prompt (middle-row), and the attention map of the same prompt without residuals (bottom-row) evaluated at different layers. Attention maps for this prompt from early (close to input) to later (close to output) layers are arranged from left to right in each row.