Is Visual Prompting the Right Setup for Knowledge Transfer in new Foundation Models?

Niclas Hergenröther [1] Antonio Orvieto [1]

Abstract

Visual Prompting (VP) has emerged as a promising technique for efficient knowledge transfer. As new foundation model families (like Mamba) get introduced and VP pipelines such as AutoVP reach greater maturity, we find a growing need for a systematic evaluation of current approaches. In this work, we assess the performance of the latest models, comparing them to earlier architectures and alternative fine-tuning methods, to better understand the progress, challenges and opportunities in the field of efficient fine-tuning under resource limitations. Towards this goal, this paper provides a concise empirical overview of the interactions among foundation model families (Attention-, Convolution-, and Mamba-based) and transfer paradigms: VP, Linear Probing (LP), and Full Finetuning (FFT). Our work builds up on previous findings by broadening the selection of evaluated models, tuning hyperparameters, and techniques. In the interest of delivering practical guidelines for the user, we also explore application of prevalent regularization techniques to boost performance in the context of VP.

1. Introduction

Efficient transfer of pretraining knowledge to new tasks and domains is an essential requirement for successful downstream applicability of foundation models. As the AI landscape continues to grow – spanning diverse architectures and design choices – so do options for fine-tuning, including new pipelines for Visual Prompting (VP), Linear Probing (LP), and Full Finetuning (FFT). For a review of these methods, please refer to Section 2. Given this rapid expansion, it is critical to systematically and regularly assess how different architectural choices interact with various transfer learning methods. In this work, we present an unbiased evaluation of VP across several prominent model architectures, analyze how performance correlates with pretraining accuracy, and discuss potential performance and compute tradeoffs when comparing to other finetuning methods on challenging tasks.

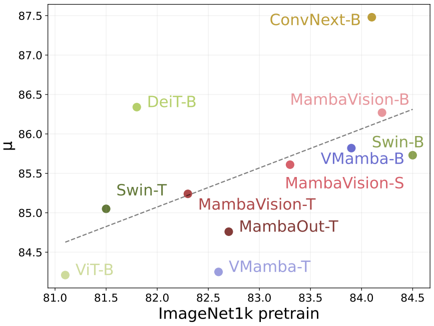

Figure 1: *ImageNet1k-pretrain and average VP top-1 accuracy over the tasks described in Section 3. We follow the recipe of Tsao et al. (2024). All models are tuned to best. Dotted line is linear regression, full results in Table 1.*

1. Compared to earlier results (Misra et al., 2024) for ILM-VP (Chen et al., 2023), we found that vision transformers and Mamba-based (Gu & Dao, 2023) models perform similarly – with only a slight edge for the former – under latest VP practice (Tsao et al., 2024). However, when selecting a model for VP, **ConvNext appears to be the best choice** from our selection (Figure 1). References and a short description for these models are presented in Section 3.
2. State-of-the-art VP practice can underperform (overfit) on data scarce tasks compared to LP and FFT – see Table 2: when access to the model is given and memory is the only constraint, LP appears to be overall the best performing paradigm (Table 4). In the FFT and especially in the LP setup, we found **Mamba to be the best choice**.
3. It is natural to ask at this point if VP performance can be improved via regularization. We found that prevalent **regularization techniques can boost generalization**, whilst still underperforming FFT. We noticed that most gains can be observed when applying regularization to the label mapping weights, as opposed to the prompt.

[1] ELLIS Institute Tübingen, Tübingen AI Center, Max Planck Institute for Intelligent Systems, Tübingen. Correspondence to: Niclas Hergenröther <niclas.hergenroether@student.unituebingen.de>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

	CIFAR10	CIFAR100	Flowers102	GTSRB	DTD	EuroSAT	Food101	OxfordPets	μ	Pretrain acc.
VMamba-T	94.89	80.53	91.67	86.75	67.17	95.72	69.25	88.07	84.25	82.60
MambaOut-T	95.07	81.36	89.58	88.27	67.73	95.70	71.26	89.14	84.76	82.70
MambaVision-T	95.44	80.89	92.55	87.74	66.26	96.56	71.56	90.91	85.24	82.30
Swin-T	94.28	80.32	92.34	88.95	66.37	96.31	72.99	88.83	85.05	81.50
MambaVision-S	95.97	82.37	90.71	86.93	66.58	96.62	74.18	91.57	85.61	83.30
MambaVision-B	96.54	83.88	91.79	88.79	65.64	96.63	**75.65**	91.22	86.27	84.20
ConvNext-B	96.42	**84.44**	**94.15**	90.28	**70.77**	96.61	75.10	**92.04**	**87.48**	84.10
VMamba-B	**96.58**	76.90	93.67	90.56	67.87	94.14	75.52	91.35	85.82	83.90
DeiT-B	96.52	83.59	93.51	**91.57**	66.54	**96.85**	72.13	90.01	86.34	81.80
ViT-B	95.52	78.18	90.74	89.54	62.13	96.80	70.84	89.97	84.21	81.10
Swin-B	95.45	81.55	93.35	86.19	68.47	96.59	74.26	89.95	85.73	84.50
Corr. ImageNet	0.49	0.31	0.35	-0.23	0.66	-0.23	0.73	0.53	0.60	

Table 1: *Top-1 accuracy after training with AutoVP using fully connected label mapping. μ denotes the average fine-tuning performance. The last column represents the ImageNet pretrain accuracy. The bottom row reports the correlation coefficient between the corresponding column and the ImageNet pretrain accuracy.*

2. Background

Benchmarking Visual Prompting. VP, first introduced as Adversarial Reprogramming (Elsayed et al., 2018), leverages the idea of utilizing a universal input perturbation to transfer a pretrained model to a new domain or task. The term Prompting originates from language modeling (Guo et al., 2017; Lester et al., 2021) and was later adapted to the vision domain (Bahng et al., 2022). VP keeps the pretrained source model entirely frozen and therefore differs from other techniques (see next paragraph) that may require internal access. Indeed, this characteristic allows finetuning in settings where the **access to model weights is not given** (Chen et al., 2021), e.g. when using a model only available via API (Tsai et al., 2020).

AutoVP, Tsao et al. (2024) introduced improvements of the VP methodology as well as a unified benchmark for VP. The introduced improvements include the possibility of resizing the prompt as well as a new Label Mapping (LM) technique: Fully Connected Label Mapping (FCLM). FCLM is realized by a fully connected layer of dimensions $C_S \times C_T$, C_S being the number of source classes and C_T the number of target classes. While there exists reserach on the combination of new foundation models and VP (Chen et al., 2023; Tsao et al., 2024; Misra et al., 2024), these works are either limited in model selection or didn't tune hyperparameters, making the results ambiguous to interpret. In this work, we evaluate a selection of ImageNet1k pretrained foundation models (Sec. 3) using AutoVP (Tsao et al., 2024) on a common set of downstream image-classification tasks and in addition to performance (Tab. 1 and Fig. 1) also investigate on the behavior of models w.r.t. the resizing of the prompt (Tab. 5 and Fig. 2).

Full fine-tuning and linear probing. Based on the improvements of AutoVP, we conduct a comparison of FFT, LP (described next) and VP on the data scarce DTD dataset (Cimpoi et al., 2014), which showed to be the most challenging task in the benchmark. We investigate a subset of the models evaluated in Tab. 1, while tuning learning rate and weight decay to ensure optimal performance for each method. LP freezes the pretrained model, whilst only replacing and training a new classifier with the idea of leveraging the existing feature representations at the penultimate layer of a model (Alain & Bengio, 2018). In contrast to full finetuning, LP drastically reduces the amount of trainable parameters as well as hardware requirements. Last, in our implementation of FFT, we account for the new number of output classes and we replaced the classifier and trained a new classifier from scratch (Chen et al., 2023). In

	VP	LP	FFT
MambaVision-B	65.64	**75.11**	**75.16**
ConvNext-B	**70.77**	72.07	74.36
VMamba-B	67.87	68.30	74.42
DeiT-B	66.54	67.77	72.93
ViT-B	62.13	63.51	69.31
Swin-B	68.47	68.40	74.79

Table 2: *Top 1 test-performance in % of various base-models on DTD using different Transfer Learning paradigms. Learning rate and weight decay are tuned. Mamba shows a clear advantage in LP.*

contrast to LP or VP, no model parameters are frozen. While this unlocks stronger results, it also requires larger compute and memory (Tab.4) compared to the other approaches. Exploring the application of LoRA (Hu et al., 2022) could also provide valuable insights and is left for future work.

Regularizing Visual Prompting. We noticed models showed signs of overfitting when being transferred to DTD using VP – as testified by Table 2, where **all models reached $\geq 99\%$ training accuracy despite suboptimal test performance compared to LP or FFT**. This poses the question on how to effectively regularize VP. This task is particularly interesting, as common regularization methods are motivated by a different learning paradigm. We investigated a set of common regularization approaches while also ablating on targeting regularizing only to a subset of the optimized parameters in VP: either prompt or Label Mapping parameters. Specifically, we evaluated performance of Dropout (Hinton et al., 2012), L1-/L2-regularization, Sharpness-aware minimization (SAM) (Foret et al., 2020) and Autoaugment (Cubuk et al., 2019).

3. Experimental setup

The tasks are selected in accordance with earlier evaluations on VP (Chen et al., 2023; Tsao et al., 2024), consisting of CIFAR10/100 (Krizhevsky et al., 2009) Flowers102 (Nilsback & Zisserman, 2008),GTSRB(Houben et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), Food101 (Bossard et al., 2014), OxfordPets (Parkhi et al., 2012). The models we use are VMamba[1] (Liu et al., 2024), MambaOut[2] (Yu & Wang, 2025), MambaVision[3] (Hatamizadeh & Kautz, 2024), SwinT[4] (Liu et al., 2021), ConvNext[4] (Liu et al., 2022), ViT[4] (Kolesnikov et al., 2021) and DeiT[5](Touvron et al., 2021). In all cases, the ImageNet (Deng et al., 2009) pretrained weights are utilized.

- Our initial set of experiments (Table 1 and Section 4.1) involves training the previously introduced models – belonging to drastically different architecture families – on the AutoVP Benchmark (Tsao et al., 2024). For detailed information regarding hyperparameter tuning, please refer to Appendix 5.1.
- Our second series of experiments (Section 4.2) entails a comparative evaluation of LP, VP, and FFT on a subset of the introduced models. These fine-tuning paradigms are compared on the DTD dataset, as it appears to be the most challenging task with limited training data.
- Our third set of experiments (Section 4.3) focuses on the regularization of VP. To assess regularization strategies on VP, we evaluate VMamba-B on the DTD Dataset (with Vanilla-VP performance obtainable from Table 1), utilizing a fixed learning rate of 0.001. Dropout and L1/L2 regularization are implemented both across the entire model or specifically targeted to either the mapping or the prompt parameters.

All runs from the first and second set of experiments are using a single Nvidia A100 GPU. The remaining experiments are utilizing either Nvidia Tesla V100 or Nvidia Quadro RTX 6000 GPUs.

4. Results

4.1. VP levels model performances

Tab. 1 presents the model performances alongside the correlation between model performance and pretraining ImageNet accuracy. In general, raw ImageNet pretraining accuracy presents itself to be a determining factor for model performance, as validated by the reported correlation coefficient. Nonetheless, certain tasks exhibit weak or even negative correlation.

ConvNext achieves the highest overall performance and dominated the majority of tasks. Yet, some tasks are more sensitive to model choice than others: while on CIFAR10 and EuroSAT performance differences are marginal, on DTD, the right selection of pretrained model is crucial.

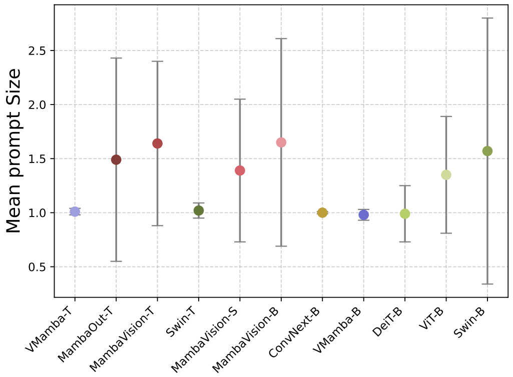

Figure 2: *Mean prompt size of the best performing run, with the initial prompt scaling being 1. Mean and standard deviation calculated over the tasks introduced in Sec. 3.*

[1] Imported from the VMamba repository (https://github.com/MzeroMiko/VMamba)
[2] Imported from the MambaOut repository (https://github.com/yuweihao/MambaOut)
[3] Imported from the MambaVision repository (https://github.com/NVlabs/MambaVision)
[4] Imported using torchvision (maintainers & contributors, 2016)
[5] Imported using timm (Wightman, 2019)

Our last investigation concerns the comparison with VP methods prior to Tsao et al. (2024), and in particular with ILM-VP (Chen et al., 2023), tested also by Misra et al. (2024). In contrast to ILM-VP, the variance in performance between different model architectures is significantly reduced. Using AutoVP on the Flowers102 dataset increases the performance of VMamba by 54.77% in comparison to ILM-VP, where it only reaches 38.9%. In comparison, ViT-B already achieves 83.1% under ILM-VP and is outperformed by VMamba under AutoVP.
Towards understanding these differences due to the VP technique, we investigate the prompt-resizing feature of AutoVP. As depicted in Fig.2, the capability to resize the prompt is not exploited uniformly across models. ConvNext, which on average performs best, retains the initial configuration, while Swin-B exhibits considerable variation in prompt size. Models differing only by size (e.g., MambaVision, VMamba) tend to have similar behavior, while Swin Transformer behaves as an outlier with Swin-T maintaining the initial prompt size and Swin-B showing variation. Detailed results can be found in Appendix 5.1.

4.2. Comparison training paradigms

	# PARAM.	GPU-MEM (GB)	EPOCHTIME (S)
VP	197,576	12.5	39.8
LP	48,175	11.2	36.1
FFT	87,601,967	72.3	49.2

Table 3: *VP refers to AutoVP with fully connected label mapping and adjustable prompt size. GPU-Mem describes the peak allocated GPU Memory. Reported Runtime is the median over all runs. All reported numbers refer to the use of VMamba-B on the DTD dataset.*

The evaluation of different finetuning techniques must consider the distinct computational requirements for training models of significant size. The precise calculation of trainable model parameters for the different approaches is contingent on numerous factors, including the selected model, input dimensions, and the number of target classes. For this comparison we assume VMamba-B being applied to the DTD dataset with 47 target classes. For VP, inputs are resized using a learnable prompt size, while for LP and FFT they are resized to the input size 224 × 224. Furthermore, while the trainable parameters of all approaches depend on the number of target classes, VP is also reliant on the input dimensions and the number of source classes in the pretrained model while LP also depends on the feature dimension of the penultimate layer. Consequently, the reported numbers need to be interpreted within the context of the specific choice of pretrained model and task. As showcased in Tab. 4, for the specific setting described above, LP proves to be the most lightweight finetuning method, with VP being only slightly more hardware demanding. By requiring only about 15% of the GPU-memory required for FFT, the field of application is drastically broadened by, in contrast to FFT, being able to e.g. run on a Nvidia Tesla V100 GPU. The results (Tab. 2) stand in contrast to previous findings (Zheng et al., 2023), indicating that, on small datasets, a smaller number of trainable parameters is beneficial and VP should be preferred over FFT. It is also noteworthy that ConvNext underperforms Mamba on FFT whilst drastically outperforming them on VP.

4.3. Regularization of VP

REFERENCE	67.87
DROPOUT	68.73
L1 ON PROMPT	69.42
L1 ON MAPPING	69.37
L2 ON PROMPT	68.84
L2 ON MAPPING	**70.31**
AUTOAUGMENT	68.65
SAM	69.31

Table 4: *AutoVP performances of VMamba-B using different regularization mechanics. First row reports the unregularized VP performance.*

Applying L1 regularization to either the prompt generation or the Label Mapping leads to almost similar enhancements. Conversely, for L2 regularization, regularizing the Label Mapping is substantially more effective than regularizing the prompt. The automatic augmentation schemes from Cubuk et al. (2019), which are derived from ImageNet, SVHN, and CIFAR10, increased performance, however to the smallest degree. VMamba also benefits from the utilization of SAM as well as from applying Dropout to the Label Mapping.

Although regularization helps in enhancing VP performance, in particular when applied to the Label Mapping, a significant discrepancy with FFT performance (Table 2) still exists, highlighting that overfitting on VP cannot be fully addressed through traditional regularization methods. Also, no single regularization method can completely close the performance gap between VMamba and ConvNext.

Conclusion

Although AutoVP has significantly enhanced the capabilities of VPs, it can underperform compared to LP and FFT in contexts with limited data availability, also in the presence of regularization. When employing VP, our results show that performance is correlated with pretraining accuracy, but depends on model architecture, with ConvNext being the best option in our setup. Future investigations might include a comparison with low-rank adapters, and comparisons of LP and VP on other data-scarce tasks.

References

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes, 2018.

Bahng, H., Jahanian, A., Sankaranarayanan, S., and Isola, P. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Chen, A., Yao, Y., Chen, P.-Y., Zhang, Y., and Liu, S. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19133–19143, 2023.

Chen, L., Fan, Y., and Ye, Y. Adversarial reprogramming of pretrained neural networks for fraud detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 2935–2939, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014. doi: 10.1109/CVPR.2014.461.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Elsayed, G. F., Goodfellow, I. J., and Sohl-Dickstein, J. Adversarial reprogramming of neural networks. *CoRR*, abs/1806.11146, 2018.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hatamizadeh, A. and Kautz, J. Mambavision: A hybrid mamba-transformer vision backbone. *arXiv preprint arXiv:2407.08083*, 2024.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7): 2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., and Zhai, X. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., and Liu, Y. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11966–11976, 2022. doi: 10.1109/CVPR52688.2022.01167.

maintainers, T. and contributors. Torchvision: Pytorch's computer vision library, 2016.

Misra, D., Gala, J., and Orvieto, A. On the low-shot transferability of [v]-mamba. *arXiv preprint arXiv:2403.10696*, 2024.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Tsai, Y.-Y., Chen, P.-Y., and Ho, T.-Y. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning*, pp. 9614–9624. PMLR, 2020.

Tsao, H.-A., Hsiung, L., Chen, P.-Y., Liu, S., and Ho, T.-Y. AutoVP: An Automated Visual Prompting Framework and Benchmark. In *The Twelfth International Conference on Learning Representations*, 2024.

Wightman, R. Pytorch image models, 2019.

Yu, W. and Wang, X. Mambaout: Do we really need mamba for vision? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zheng, Y., Feng, X., Xia, Z., Jiang, X., Demontis, A., Pintor, M., Biggio, B., and Roli, F. Why adversarial reprogramming works, when it fails, and how to tell the difference. *Information Sciences*, 632:130–143, 2023. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2023.02.086.

5. Appendix

5.1. Benchmarking Visual Prompting

The Experiments from 4.1 followed the AutoVP recipe (Tsao et al., 2024) with:

- Learning rates of [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3]
- 2 different seeds and no regularization applied.
- Initial image-scale of 1.0 with prompt scaling enabled
- 50 epochs

The reported accuracy is the maximum test accuracy over the run.

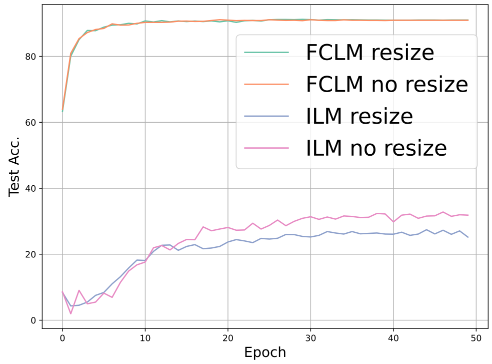

Figure 3: *Flowers102 performance of VMamba-T for combinations of ILM and FCLM with and without enabled resizing*

As depicted in Fig. 3, the enhanced performance of VMamba can be attributed to FCLM rather than to the resizable prompt. Surprisingly, for the specific runs, prompt resizing even degraded the performance. Runs comparing the different Mapping approaches were conducted with learning rates [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3] with one seed.

	CIFAR10	CIFAR100	Flowers102	GTSRB	DTD	EuroSAT	Food101	OxfordPets	μ	σ
VMamba-T	0.990	0.999	0.996	1.047	0.999	1.083	1.000	0.999	1.014	0.031
MambaOut-T	1.000	0.999	0.999	3.674	0.997	0.691	1.254	2.275	1.486	0.938
MambaVision-T	0.995	0.998	1.401	3.111	2.195	0.927	1.182	2.342	1.644	0.756
SwinT-T	0.986	1.001	0.999	0.995	0.999	0.950	1.000	1.196	1.016	0.070
MambaVision-S	0.927	1.006	2.930	1.105	0.997	1.006	1.238	1.942	1.394	0.656
MambaVision-B	1.003	1.006	1.002	3.912	2.362	0.994	1.386	1.574	1.655	0.960
ConvNext-B	0.994	1.001	0.999	0.991	0.998	0.985	1.000	1.001	0.996	0.005
VMamba-B	1.004	1.021	0.997	0.863	0.999	0.927	1.000	0.998	0.976	0.050
DeiT-B	1.011	1.000	0.998	0.792	1.000	0.569	1.000	1.562	0.991	0.261
ViT-B	1.004	0.856	1.090	0.998	1.349	1.000	2.332	2.190	1.352	0.542
Swin-B	1.003	0.997	1.001	4.798	0.999	1.003	1.214	1.542	1.569	1.233

Table 5: *Prompt scaling of the best run for the respective model-task combination*

Tab. 5 provides detailed results regarding the Prompt resizing for different Model-task combinations. The reported number is the prompt size after the last training epoch w.r.t. the initial instantiation.
An interesting observation is that while some tasks generally lead to a stronger resizing (e.g. GTSRB results in an average prompt size of 2.026 times the initial calibration), 5 out of the 11 models observed also reduce the prompt size.

5.2. Comparing Knowledge Transfer Methods

LP and FFT were implemented by adapting https://github.com/kuangliu/pytorch-cifar. The results for VP were obtained using the setup described in 5.1.

The models were all evaluated on FFT and LP using the combinations of the learning rates [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3] and weight decays [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3]. All models were trained for 200 epochs on one seed.

`RandomRotation((10))`, `Colorjitter(brightness=0.05,contrast=0.05,saturation=0.05)` and `RandomAffine(degrees=5,translate=(0.1,0.1),scale=(0.9,1.1))` were used as input perturbations.

For LP and FFT, the linear layer of the classification head was replaced to account for the correct number of output classes. For VP, the rest of the model was frozen and only the new classifier was trained.

5.3. Regularizing Visual Prompting

The Experiments from 4.3 used the best performing learning rate of 0.001.

- Dropout was applied with p-values of [0.05, 0.1, 0.2, 0.4] on the Label Mapping.
- L1 and L2 regularization were both run with λ-values of [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3] on either the prompt or the Label Mapping.
- Autoaugment was used by applying the precomputed schemes for ImageNet, SVHN and CIFAR10.
- For SAM, ρ of [0.001, 0.003, 0.005, 0.008 and 0.01] were used.