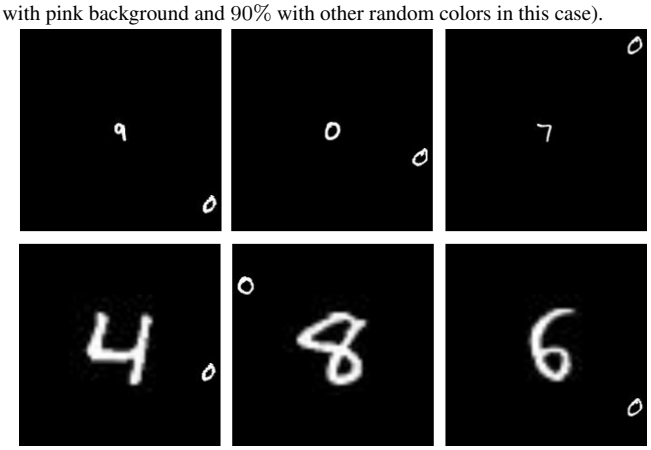
Supplementary for Black-Box Visual Prompting for Robust Transfer Learning

A. Experimental Setting

A.1. Datasets

Synthetic Datasets Our BlackVIP generates the input-dependent image-size visual prompt which covers the whole image region, so we expect that this flexible prompt design can improve some kind of robustness as well as general recognition capability: (1) To evaluate the robustness on distribution shift (i.e., domain generalization), we consider Biased-MNIST [1] dataset. (2) To evaluate the robustness on adversarial noise and location-agnostic recognition capacity, we create a variant of the MNIST dataset called Loc-MNIST. Examples of these two datasets are provided in Figure 1.

(a) Examples of $y = 7$ subset in Biased-MNIST [1] with $\rho = 0.9$. (Top) the train set is constructed with the spurious correlation between the background color and digit class (e.g., $y = 7$ occurs 90% with pink background and 10% with other random colors in this case). (Bottom) the test set is constructed with a reversed correlation to that of the train set (e.g., $y = 7$ occurs 10% with pink background and 90% with other random colors in this case).



(b) Examples of Loc-MNIST dataset. The real digit from MNIST is located in the outer area, while the fake digit from another random MNIST image is placed in the center of the image. (Top) the case where the size ratio of the real digit to the fake digit is 1:1, and (Bottom) 1:4.

Figure 1. Examples of two synthetic datasets. (a) Biased MNIST and (b) Loc-MNIST.

Biased MNIST is a modified version of MNIST [15] where the biases reside in the background colors of the images of each digit. At train time, each digit has a unique preassigned background color that strongly correlates with the label. The degree of correlation is determined by the value $\rho \in [0, 1]$, such that $(100 \times \rho)\%$ of the images that belong to the same digit have the preassigned color of that digit as their background color, and the rest are uniformly assigned to have any of the other colors as their background color. At test time, we reverse the ratio so that $(100 \times (1-\rho))\%$ of the images now have the preassigned color as their background color and vice versa to evaluate the model's dependency on superficial features such as the color of the background that a digit is located on. We prepare the following two environments 1) easy: $\rho = 0.8$ and 2) hard: $\rho = 0.9$.

On the given black blank image with 224×224 resolution, i.e., zero's array, **Loc-MNIST** puts an original target digit image from MNIST that has 28×28 resolution on the edge-side (e.g., 0∼27 or 196∼223 for one of vertical or horizontal side and 0∼223 for another side) and puts a random fake digit (also from the MNIST dataset) on the center. The location of the target digit in the edge and the class of fake digit are chosen randomly with uniform probability. A synthetic image is created one by one for each original MNIST image. We prepare the following two environments 1) easy: the scale of the target and the fake digit is the same, i.e., 1:1, and 2) hard: the fake digit is four times larger than the original digit, i.e., 1:4.

For consistency, we perform the experiments on these two datasets with a few-shot evaluation protocol. To construct a train set, we randomly sample a subset (K-shot) of the created images for each class and use the whole test set.

Datasets To extensively evaluate the effectiveness of our proposed method and baseline approaches, we measure performance across the following 14 datasets that are widely used for transfer learning benchmark: Caltech101 [9], OxfordPets [20], StanfordCars [14], Flowers102 [19], Food101 [3], FGVCAircraft [16], SUN397 [26], DTD [6], SVHN [18], EuroSAT [11], Resisc45 [4], CLEVR [13], UCF101 [22], and ImageNet (IN) [7]. Note that these 14 datasets cover diverse visual domains, and they require understanding various visual semantics like scenes, actions,

fine-grained categories, textures, satellite imagery, digits, the number of objects, and the recognition of generic objects.

Following the protocol in [27,28], we conduct a few-shot evaluation for all datasets: 16-shot for the train set, 4-shot for the validation set, and the whole test set. We use the few-shot split by [28] for each dataset those are also used in [28], while for Resisc45 and CLEVR, we randomly select the 16-shot and 4-shot samples for training and validation dataset, respectively.

A.2. Backbone Model

In this work, we aim at the robust adaptation of pre-trained models on diverse downstream tasks. For these pre-trained models, all experiments in this paper are done with the off-the-shelf vision-language model CLIP [21], and we adopt the ViT-B/16 for image encoder backbone architecture by default. During the adaptation (training) phase, the entire components of the pre-trained model are frozen without any architectural modification, and we only manage and optimize the learnable module Coordinator from the outside of the pre-trained model.

While input space visual prompting allows it to be applied to not only VLM, but also any other vision models like CNNs and ViTs, it requires the user to define the output space mapping, which maps the output prediction category set of a pre-trained task to a new downstream category set [2, 8, 25]. This is another non-trivial problem. Therefore, we limit our focus to only the VLM that can dynamically build the task-specific head from manual text template [12, 21] so that free from defining output space mapping.

A.3. Baseline Methods

CLIP Zero-Shot (ZS) CLIP [21] is one of the most popular vision-language zero-shot models that is widely exploited for classification, detection, segmentation, and other vision or vision-language tasks. Based on its well-aligned vision-language joint embedding space, the zero-shot classification can be performed with a manual text prompt (also called template) of each pre-defined class category. In this paper, we are mainly aiming to improve the CLIP's strong zero-shot performance in the few-shot adaptation setting.

BAR Black-Box Adversarial Reprogramming (BAR) [25] was proposed for efficient transfer learning of pre-trained model to the medical image domain. Different from the previous works on Adversarial Reprogramming (AR), BAR exploits the perturbation-vulnerability of neural networks for *adaptation* purpose rather than attack. By optimizing the frame-shaped learnable program, which embeds a downstream target image inside of that, BAR steers the ImageNet pre-trained model to classify the specialized medical images. Moreover, BAR adopts the zeroth-order optimizer (ZOO), Randomized Gradient-Free (RGF) [17] minimization algorithm for black-box transfer learning to broaden its applications.

When the resolution of the downstream input image is over that of the pre-training phase, Tsai et al. [25] set the embedded target image size for 64×64 resolution in the 299×299-size learnable program by default. However, we observe that such a heavy-pad thin-image design of prompt degrade the performance significantly, so we tune the resolution of the embedded image and set 194×194.

VP Similarly, Visual Prompting (VP) aims at adapting a pre-trained model to downstream tasks via learning input space visual prompts. Among some candidates for prompt designs, Bahng et al. [2] adopt the padding-style prompt so that realized prompts look like the frame-shape program of ARs. VP learns a universal visual prompt per each downstream task, and it just adds to all of the images in a task. Unlike the AR methods or our BlackVIP, the range of prompted images is unbounded. Following [2], we use the padding-style prompt, which is 30-pixel sized for each side by default.

While VP optimizes the parameters in the input space, it relies on a first-order optimization algorithm that uses the true gradient of entire model parameters, and we establish the performance of VP as an upper bound for other input space black-box optimization approaches, including BlackVIP. Additionally, by replacing the first-order algorithm with zeroth-order counterparts, we build two new baselines **VP w/ SPSA** and **VP w/ SPSA-GC** on our extensive experiments. These two methods confirm the effectiveness of our new components *Coordinator* and SPSA-GC.

Discussion Although BAR, VP, and BlackVIP share the generic goal: efficient transfer learning of pre-trained models via input-space optimization, there are several significant differences. (1) We propose a novel prompt design that is automatically formed in an input-dependent manner rather than the frame-shaped manual design of the input-independent prompt (or program) of VP (or BAR). (2) While VP relies on first-order algorithms and BAR adopts the RGF, we utilize the new variants of SPSA [23], SPSA-GC, which is enhanced with a proper modification in the parameter update rule. (3) Contrary to the medical imaging-only validation in BAR, based on the above two technical difference, BlackVIP successfully adapt the pre-trained model to diverse data domains (described in Section B.1.).

A.4. Implementation Details

Architecture For the fixed text prompt design of each dataset those are shared across all baseline methods and BlackVIP, we use the same templates provided by [2] for

SVHN, CLEVR, and Resisc45, and [28] for remaining 11 datasets. For the frozen feature extractor (encoder) part of our *Coodinator*, we use the ImageNet pre-trained `vit-mae-base` checkpoint[1] from the HuggingFace. The output shape of the encoder is $N \times 768$, where N is the number of instances in the batch. We design the decoder based on *depth-wise separable convolution* (DSC) layer [5] for parameter efficiency. Specifically, we build a block of [`NORM-ACT-CONV`] and stack it five times. The `NORM` and `ACT` denote Batch Normalization and Gaussian Error Linear Unit, respectively. The `CONV` operation of the first four blocks is DSC, and the last one is a standard convolutional layer. Our implementation code is enclosed in `.zip` file.

To satisfy a fully convolutional design without loss of expressiveness, tensors that are fed into the decoder must be shaped in a 3D feature map. For this, we additionally govern a task-specific single continuous vector ϕ_t (called *prompt trigger vector*), which is concatenated with the output feature vector of encoder leading the appropriate size of 1d vector for reshaping to 3d tensor. In this work, we set the dimension of the prompt trigger vector to 800, resulting in 1568 dimensions of concatenated vector that can be reshaped to $32 \times 7 \times 7$ shaped 3D tensor. The prompt trigger is shared across all instances for a given task.

Optimization and other configurations For a stable approximation of gradient in practice, ZOO algorithms repeat the gradient estimation step for several times and use the mean of those estimates as a final approximation of the gradient. Usually, the approximation quality is proportional to the number of these repeats. We set this repeat as five times for all baselines that use ZOO.

Besides the learning rate and learning rate schedule parameters, ZOO algorithms have some additional algorithm-specific hyperparameters needed to be tuned. For RGF, these are the standard deviation of a random gaussian vector and a smoothing parameter, and for SPSA, these are the perturbation magnitude and its decaying factor. We provide the search range of each hyperparameter in Table 1. The search range for algorithm-specific parameters is based on the proposal of authors of SPSA [24] and BAR [25]. Moreover, among the valid perturbation distributions of SPSA, we adopt the Segmented Uniform $[-1.0, -0.5] \cup [0.5, 1.0]$.

The learning objective is a cross-entropy loss for VP and BlackVIP and focal loss for BAR (following [25]). For all black-box approaches, the batch size is set to 128 across all datasets. Except for the SUN397 (1,000), StanfordCars (2,500), and ImageNet (500), we optimize all methods during 5,000 epochs for convergence. Note that the input space visual prompting with first-order algorithm already requires sufficiently large iterations, e.g., 1,000 epoch [2] with full dataset, and ZOO demands much more iterations due to the lack of gradient information.

[1] https://huggingface.co/docs/transformers/model_doc/vit_mae

A.5. Hyperparameter Sweep

In this section, we provide the hyperparameter search range of each algorithm, summarized in Table 1.

Table 1. Hyperparameter sweep. Large LR (learning rate) of BAR and VP is based on [2] to directly optimize pixel values rather than the neural network's weights. PM denotes perturbation scale, c_i.

Hyperparameter	Algorithm	Search Range
initial LR	BAR, VP	{40.0, 20.0, 10.0, 5.0, 1.0}
initial LR (a_1)	BlackVIP	{1.0, 0.1, 0.01, 0.005}
min LR	BAR	{0.1, 0.01, 0.001}
decaying step	BAR	{0.9, 0.5, 0.1}
LR decaying factor	VP, BlackVIP	{0.6, 0.5, 0.4, 0.3}
initial PM (c_1)	BlackVIP	{0.01, 0.005, 0.001}
PM decaying factor	BlackVIP	{0.2, 0.1}
std. of perturbation	BAR	{1.0, 0.5}
smoothing	BAR	{0.1, 0.01, 0.001}
gradient smoothing	VP, BlackVIP	{0.9, 0.7, 0.5, 0.3}

B. Detail Description of *Coodinator*

On the transfer learning of a pre-trained model which provides no accessibility about any architectural information or actual model parameters, BlackVIP treats this situation with two novel mechanisms: (1) parameter-efficient instance-aware prompt generation network, and (2) stable zeroth-order optimization algorithm that is based on SPSA [23]. In this section, we provide a detailed description of the first component, Coordinator.

Different from existing works on visual prompting, we reparameterize the input space visual prompt ϕ as a neural network, *Coordinator* $h_{\boldsymbol{\phi}}(\cdot)$ that generates an input-dependent visual prompt $h_{\boldsymbol{\phi}}(x)$. Coordinator is composed with encoder $f(\cdot)$, decoder $g_{\phi_d}(\cdot)$ and task-specific learnable vector ϕ_t. The encoder is used for extracting instance-specific latent feature vector $z_x = f(x)$ contributing to the construction of the optimal input space visual prompt for each instance. Because our goal in this work is the broad utilization of pre-trained models on diverse downstream tasks, we adopt a pre-trained encoder network optimized by a self-supervised learning objective, not by a supervised learning objective or scratch network. Specifically, we use the ViT-B/16 weights from the *Masked AutoEncoding* pre-training [10]. We present the grounds for using the self-supervised learning encoder in the main paper, refer to Sec. 3. During the training phase, this pre-trained encoder part is frozen (not updated) and just acts as a feature extractor. Then, the instance-specific feature vector from the encoder is conveyed to the decoder for a prompt generation.

Prompt decoder $g_{\phi_d}(\cdot)$ is a lightweight convolutional neural network, which has learnable parameters **less than 10K** by default. Note that the generated prompt has the same shape as the input image, so our prompt covers the entire region of the image, unlike previous visual prompting and reprogramming works applied to the partial region of the image by human-designed.

In addition to the feature vector from the fixed encoder, the decoder also incorporates an additional input which is shared for all instances across the current dataset. The so-called *prompt trigger vector* ϕ_t is a continuous vector that also contributes to the design of a visual prompt by collaborating with the instance-specific rich feature vector from the encoder. By introducing this prompt trigger vector, the decoder of the Coordinator can enjoy additional information to generate more proper prompts for a given task. Besides, it helps to build the 3D feature map for the decoder's input, which is necessary for designing a parameter-efficient fully convolutional decoder network.

References

[1] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR, 2020. 1

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 2, 3

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing. 1

[4] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 1

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 3

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

[8] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *arXiv preprint arXiv:1806.11146*, 2018. 2

[9] Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. 1

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 3

[11] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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. 1

[12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

[13] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 1

[14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 1

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[16] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1

[17] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, apr 2017. 2

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 1

[19] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 1

[20] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. 1

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[22] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1

[23] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992. 2, 3

[24] James C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., USA, 1 edition, 2003. 3

[25] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning*, pages 9614–9624. PMLR, 2020. 2, 3

[26] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 1

[27] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[28] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 2, 3