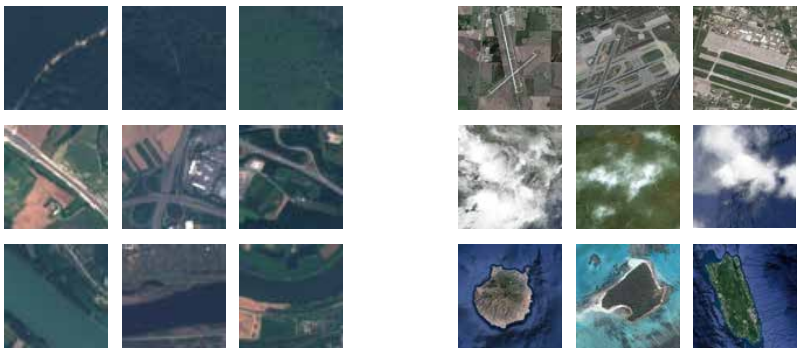
Supplementary Materials of
Generating Instance-level Prompts for Rehearsal-free Continual Learning

A. Testing Benchmarks

In this paper, we utilize seven datasets with varying levels of domain similarity to ImageNet [21, 17]. Table 5 summarizes the used datasets, number of classes, number of tasks, number of training, validation, and test images, and domain similarity to ImageNet on ViT. Furthermore, we introduce two benchmarks with a longer task sequence that includes a relatively large number of classes to validate the superior performance of DAP on larger or/and longer benchmarks. First of all, for all benchmarks, we split and utilize 20% of the training set as a validation set for the hyperparameter searches of the comparing methods and DAP. Also, to estimate the domain similarity, we follow the paper [17] which calculates Earth Mover's Distance between two domains. Refer to [17] for the technical details to estimate the domain similarity of nine datasets to ImageNet.

Dataset	# Classes	# Tasks	Train	Validation	Test	Domain Similarity [17]
Split CIFAR-100	100	10	40,000	10,000	10,000	0.491
Split Pets	35	7	2,774	706	3,469	0.470
Split EuroSAT	10	5	16,200	5,400	5,400	0.444
Split RESISC45	45	9	18,900	6,300	6,300	0.432
Split ISIC	6	3	5,940	1,980	1,980	0.409
Split ChestX	6	2	7,252	2,417	2,417	0.390
Split CropDiseases	35	7	34,260	8,566	10,692	0.386
Split ImageNet-R	200	10	18,000	6,000	6,000	0.577
Split DomainNet	345	15/69	96,724	24,182	52,041	0.600

Table 5. Specifications of the various CL benchmarks evaluated.



EuroSAT RESISC45

Figure 6. Image samples of the aerial domain, EuroSAT and RESISC45. Each row displays three samples from the same class.

Here, a description of each benchmark is provided below:

- **Natural Domain**

  1) *Split CIFAR-100*: It is built by dividing the original CIFAR-100 [12] into 10 tasks with 10 disjoint classes each.

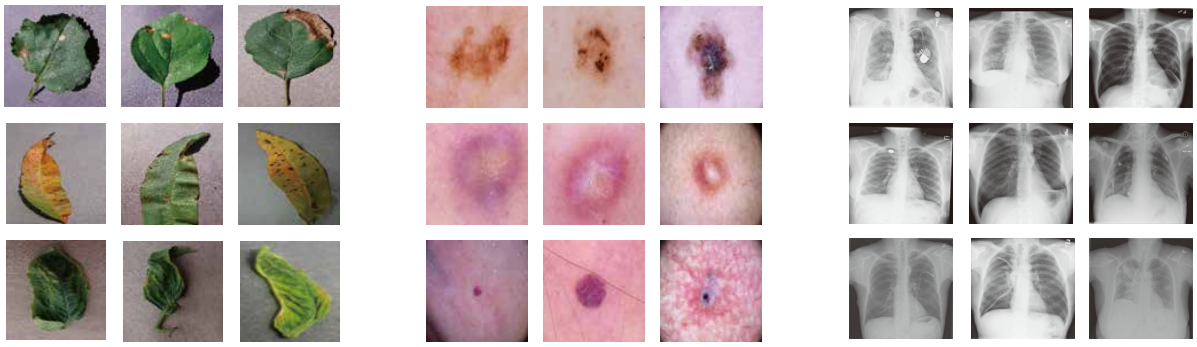


Figure 7. Image samples of the medical domain, CropDiseases, ISIC, and ChestX. Each row displays three samples from the same class.

2) *Split Pets*: Original Oxford-IIIT Pet data [18] consists of 37 pet categories (dogs and cats) with roughly 200 images per category. In the case of Oxford-IIIT Pet, because the number of data per category is similar, we drop the last two categories and configure *Split Pets* containing 7 tasks of 35 categories.

- **Aerial Domain**: The two aerial datasets are color images of natural scenes but without perspective distortion. To help understand the visual characteristics of the aerial domain, we provide representative examples of EuroSAT and RESISC45 in Figure 6.

  3) *Split EuroSAT*: EuroSAT [8] is a collection of satellite images of the landscapes based on Sentinel-2 satellite images [15]. It contains categories such as forest, highway, river and so on. *Split EuroSAT* is built by splitting the original 10 classes into 5 tasks of 2 disjoint classes each.

  4) *Split RESISC45*: RESISC45 [3] is originally built for a remote sensing image scene classification task It contains 45 scene classes, each class having 700 images, such as airport, cloud, island, and so on. *Split RESISC45* is built by splitting the 45 classes into 9 tasks of 5 disjoint classes each.

- **Medial Domain**: To help understand the visual characteristics of the medical domain, we provide representative examples of CropDiseases, ISIC, and ChestX in Figure 7.

  5) *Split CropDiseases*: CropDiseases [16] is a collection of diseased plant images, which contains natural images but is specialized in the medical and agriculture industries. The Cropdiseases dataset originally contains 38 categories. However, a class imbalance among 38 categories is severe. To relax this imbalance and facilitate task splitting, we drop the three categories with the smallest number of data in order. Therefore, *Split CropDiseases* is built by splitting the 35 classes into 7 tasks of 5 disjoint classes each.

  6) *Split ISIC*: ISIC2018 [4] is dermoscopy images of human skin lesions of 7 categories, which no longer represent natural images. This dataset also has a fairly severe class imbalance, thus we drop a single category with the smallest number of data. Then, *Split ISIC* is built by splitting the 6 classes into 3 tasks of 2 disjoint classes each.

  7) *Split ChestX*: ChestX [23] is a collection of X-Ray images of the diseased human chest. ChestX consists of gray images of 8 categories. However, considering the inherent difficulty and severe class imbalance of this dataset, we split it into 2 tasks of 3 disjoint classes by dropping the two categories with the largest and smallest number of data.

- **Large Benchmarks**: Moreover, we include two benchmarks that contain a substantial number of classes to demonstrate the robustness of DAP in handling large benchmarks.

  8) *Split ImageNet-R*: ImageNet-R [9] is a collection of images of 200 ImageNet classes. It contains a type of art, graffiti, origami, paintings, sketches, etc. As depicted in Table 5, this benchmark is derived from the 200 original ImageNet classes that are used for pre-training ViT. As a result, its domain similarity to ImageNet is considerably high. We include this benchmark not to evaluate domain scalability, but rather to assess dataset size scalability. *Split ImageNet-R* is created by dividing the 200 classes into 10 tasks, with each task containing 20 disjoint classes.

  9) *Split DomainNet*: DomainNet [19] is a dataset composed of images from 6 distinct types, consisting of a total of 345 categories. We utilize only real-type images to build *Split DomainNet*. To assess robustness in large classes and

long horizons, *Split DomainNet* is utilized in two experiments where 345 classes are split into 15 or 69 tasks, with each tack containing 23 or 5 disjoint classes, respectively.

B. Details of Comparing Methods

To verify the relative effectiveness of all methods, we also contain **FT-seq**, the naive sequential training (i.e, lower-bound), and **Upper-bound**, the supervised joint-finetuning on the data of all tasks.

EWC [11] is one of the representative algorithms in continual learning to avoid catastrophic forgetting. This method regularizes the weights of the model based on its fisher information. For a fair comparison, the weights of the model start from the weights of the ImageNet pre-trained model. **LwF** [13] is the algorithm that utilizes the distillation loss for preventing catastrophic forgetting. This method is also well-known as the baseline in continual learning like EWC, and we set the model to start from ImageNet pre-trained weights for a fair comparison. **L2P** [26] firstly proposes a prompt-based method in continual learning. It focuses on the shared prompt pool for adapting incoming sequential tasks using a pre-trained model, which is the same pre-trained model in our method for a fair comparison. **DualPrompt** [25] proposes another prompt-based method. Unlike L2P, this method maintains two types of prompts with different objectives: namely task-invariant and task-agnostic. Currently, it is the state-of-the-art prompt-based method in continual learning. We use the same pre-trained model for a fair comparison. Also, because the main method of DualPrompt is prefix tuning, all the reported results of DualPrompt are obtained with prefix tuning.

C. Hyperparameter Search Space of L2P and DualPrompt

Because DAP does not utilize the prompt pool, it is relatively less sensitive to prompt-related hyperparameters than L2P and DualPrompt. Thus, we train DAP using consistent hyperparameters throughout all benchmarks, regardless of the domain similarity to ImageNet. However, in the case of L2P and DualPrompt, which utilize the prompt pool, their performance largely relies on the choice of prompt pool size, selection size, and prompt length. Therefore, for a fair comparison of various benchmarks, we create the hyperparameter search space based on the prompt-related hyperparameters proposed in L2P and DualPropmt as follows:

1. L2P
    - Prompt pool size: [10, 20]
    - Prompt top K: [1, 4]
    - Prompt length: [5, 10, 20]
    - Diversifying prompt-selection: [True, False]
2. DualPrompt
    - G-prompt length: [5, 10, 20]
    - E-prompt pool size: [10, 20]
    - E-prompt top K: [1, 4]
    - E-prompt length: [5, 10, 20]

In the search space, we conduct a grid search to find the set of prompt-related hyperparameters showing the best performance on each benchmark of various domains. Then, we report the obtained best performance in Table 1 as L2P† and DualPrompt†. Here, we summarize the set of prompt-related hyperparameters used to obtain the results in Table 1 as follows:

1. L2P
    - *Split Pets*, *Split EuroSAT*, *Split RESISC45*
        - Prompt pool size: 10
        - Prompt top K: 1
        - Prompt length: 10
        - Diversifying prompt-selection: True
    - *Split CropDiseases*, *Split ISIC*

- Prompt pool size: 10
- Prompt top K: 1
- Prompt length: 20
- Diversifying prompt-selection: True

- *Split ChestX*
  - Prompt pool size: 10
  - Prompt top K: 4
  - Prompt length: 10
  - Diversifying prompt-selection: True

2. DualPrompt

- *Split Pets*
  - G-prompt length: 5
  - E-prompt pool size: 10
  - E-prompt top K: 1
  - E-prompt length: 5
- *Split EuroSAT*, *Split RESISC45*, *Split CropDiseases*
  - G-prompt length: 20
  - E-prompt pool size: 10
  - E-prompt top K: 1
  - E-prompt length: 20
- *Split ISIC*
  - G-prompt length: 20
  - E-prompt pool size: 20
  - E-prompt top K: 4
  - E-prompt length: 10
- *Split ChestX*
  - G-prompt length: 20
  - E-prompt pool size: 10
  - E-prompt top K: 4
  - E-prompt length: 20

D. Evaluation Metrics

- **Average accuracy [14].** It is the average accuracy of all the tasks after the model is trained on the last task T, and it is commonly utilized in the continual learning community. It can be formulated as follows:

$$\text{Avg Acc} = a_T \quad \text{where} \quad a_i = \frac{1}{i}\sum_{j=1}^{i} a_{i,j},$$

  where $a_{i,j}$ is the accuracy evaluated on the test set of the j-th task when model is trained til i-th task.

- **Forgetting measure [1].** It can be defined as the difference between the maximum knowledge from the previous tasks and the knowledge in the current task. As such, we can calculate the estimate of how much the model forgot the knowledge of the previous task j when given current task k ($k > j$), and the metric can be formulated as follows:

$$\text{Forgetting} = \frac{1}{T-1}\sum_{j=1}^{T-1} f_j^T \quad \text{where} \quad f_j^k = \max_{l\in\{1,2,..,k-1\}} a_{l,j} - a_{k,j}.$$

- **Learning accuracy [20].** It focuses on how much the model can learn new knowledge of new tasks. As such, it can be calculated as the average accuracy of each task right after the model is trained on the incoming tasks, and it can be formulated as Lrn Acc $= \frac{1}{T}\sum_{j=1}^{T} a_{j,j}$.
- **Total average accuracy [5, 24].** To validate the performance of the method on various datasets, we take an average of Avg Acc of all domains. It can be formulated as Tot Avg Acc $= \frac{1}{|\mathcal{D}|} a_{T_i}$, where T_i is the number of tasks of the domain $\mathcal{D}_i$. Total Forgetting and Total Lrn Acc can be defined in the same way using each own forgetting and learning accuracy. In this paper, we use $|\mathcal{D}| = 7$.

E. Additional Experiments

Method	Split CIFAR-100			Split EuroSAT			Split CropDiseases		
	Avg. Acc (↑)	Forgetting (↓)	Lrn. Acc (↑)	Avg. Acc (↑)	Forgetting (↓)	Lrn. Acc (↑)	Avg. Acc (↑)	Forgetting (↓)	Lrn. Acc (↑)
L2P	81.09±1.15	9.12±0.50	89.30±0.71	37.85±5.30	54.35±9.54	81.36±2.35	53.40±3.54	28.76±2.83	80.77±2.12
DualPrompt	83.25±1.87	**7.72±0.88**	89.61±0.58	69.74±1.05	20.53±3.20	**86.15±1.49**	76.31±1.88	10.00±2.69	83.84±1.05
DAP	**83.26±1.37**	8.27±1.30	**90.71±1.58**	**72.32±2.79**	**11.56±5.87**	82.44±4.80	**82.70±2.74**	**7.97±1.94**	**87.50±3.34**
Matching Acc of DAP	71.86±3.20			68.40±0.91			73.16±0.73		

Table 6. Results on class-incremental learning in an instance-wise setup and matching accuracy on same benchmarks.

In accordance with the evaluation setups of L2P and DualPrompt's official code[1], we assess performance in both batch-wise and instance-wise prompt setups. The main setup in L2P and DualPrompt is batch-wise, but we also examine the instance-wise setup. After analyzing the official code of L2P and DualPrompt, we note that the key difference between the two setups lies in the selection of prompts for instances in a batch. In the former, a single prompt is chosen for the entire batch, while in the latter, prompts are selected on a per-instance basis.

As shown in Table 6 above, in the instance-wise setup, the performance of L2P, DualPrompt, and DAP becomes lower compared to the batch-wise setup. This can be attributed primarily to imprecise prompt selection or generation. Essentially, as the number of tasks increases, L2P tends to choose a prompt that is different from the one required to make an accurate prediction for each instance among the shared prompts. For both DualPrompt and DAP, it becomes challenging to estimate the correct task-specific key. As demonstrated in Table 6, DAP exhibits superior performance compared to the other methods, despite not having a high matching accuracy. This superiority can be attributed to several factors. Firstly, the instance-level prompt generated by `MLP` with the transposed input provides fine-grained instructions that are tailored to the relationship between each input patch, regardless of task information. That is, the encoded information within the prompts is still beneficial for prediction. Secondly, the task-specific key is estimated by conditioning on the input instance features. This implies that even if the estimated task-specific key is incorrect, it would still be the one with the highest similarity to the ground-truth task. As a result, the adaptive prompt that is generated is likely to contain useful instructions for prediction.

In the case of an instance-wise setup, it requires more hyperparameter searches to find a solid set of hyperparameters. It showed stable performance in terms of forgetting when appending prompts into the first half of layers as DualPrompt suggested. Also, applying layer normalization not only at the beginning of prompt generation but also within intermediate prompts helped enhance learning accuracy (Lrn Acc).

F. Variation in the Number of Training Epochs on MLP

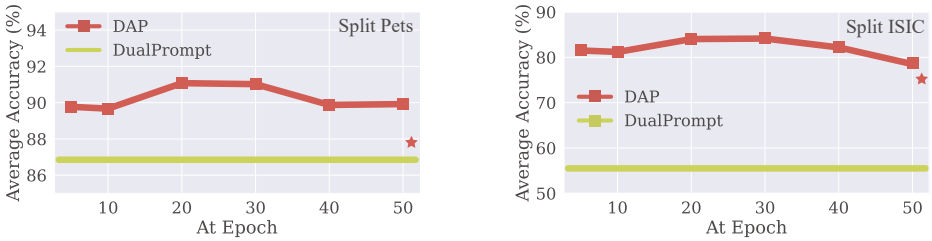


Figure 8. Ablation study on the number of training epochs before freezing the `MLP` layer and when the `MLP` keeps updated till the end of training (⋆). ⋆: no freezing on `MLP`.

[1] https://github.com/google-research/l2p/tree/main/configs

To mitigate catastrophic forgetting resulting from a distribution shift, we decide to freeze the MLP layer during training. The objective of the MLP layer is to extract instance-level domain knowledge from target data, regardless of the task at hand. Figure 8 depicts the performance of DAP with varying numbers of training epochs on MLP on two benchmarks with different domain similarity: Split Pet, which is close to ImageNet, and Split ISIC, which is distant enough. The number of training epochs per task in Split Pet and Split ISIC is 30. Our observations indicate that the performance does not vary much, even if we do not freeze the MLP layer precisely at the end of the first task.

Moreover, we confirm that DAP shows better performance than DualPrompt even when the MLP layer is not frozen till the end of training. See the red asterisk in Figure 8 compared to the green line. Specifically, Split ISIC, with relatively lower domain similarity, experiences a drop of 9.85% due to the MLP layer's continuous training to steer the pre-trained representation space. In the case of Split Pet, which has high domain similarity with ImageNet, the absolute performance drop is very small (3.36%), as the prompt needed to encode from the MLP layer is general enough to cover the benchmark. As a result, in both cases, DAP still outperforms DualPrompt when the MLP layer is trained until the end.

Avg Acc (↑)	At epoch				Avg Acc (↑)	At epoch			
Split Pets	10	20	30	40	Split ISIC	10	20	30	40
Original	89.66±0.75	91.07±0.56	91.02±0.44	89.87±3.53	Original	81.22±1.67	84.07±2.11	84.18±2.54	82.21±4.55
Shuffled class order	89.48±0.88	91.08±0.60	91.11±0.59	89.80±3.33	Shuffled class order	82.01±0.98	84.99±2.78	85.34±2.27	82.90±4.18

Table 7. Additional analysis to confirm the consistent trend on randomly shuffled class order.

In addition, we test whether this trend is maintained on benchmarks built on top of randomly shuffled class order. As shown in Table 7, we can observe consistent performance on both Split Pets and Split ISIC with the randomly shuffled class order. This finding demonstrates that our method is robust to variations in class order. Still, we observe that performance does not vary much even if we do not freeze the MLP layer exactly at the end of the first task.

G. Large number of Classes and Longer Task Sequences

Benchmark	L2P			DualPrompt			**DAP (Ours)**		
	Avg Acc (↑)	Forgetting (↓)	Lrn Acc (↑)	Avg Acc (↑)	Forgetting(↓)	Lrn Acc (↑)	Avg Acc (↑)	Forgetting(↓)	Lrn Acc (↑)
(A) Split ImageNet-R (10 tasks)	60.98±0.70	9.93±0.43	69.23±0.78	68.97±2.87	4.66±2.15	72.85±2.27	**70.12±2.24**	**2.90±2.70**	**73.24±2.81**
(B) Split DomainNet (15 tasks)	80.67±0.85	5.33±0.87	85.14±0.99	81.89±0.63	5.21±1.17	87.27±1.80	**83.51±1.07**	**5.30±0.52**	**88.77±0.79**
(C) Split DomainNet (69 tasks)	77.28±0.80	9.70±0.72	86.59±0.74	79.44±1.12	7.91±0.60	87.35±1.03	**83.36±0.81**	**6.75±1.72**	**90.50±0.74**

Table 8. Results on benchmarks with a large number of classes and longer task sequences.

To expand the impact of DAP, we evaluate its performance on benchmarks that entail a large number of classes and longer task sequences. Initially, this study primarily focuses on proposing a domain-adaptive prompt-based CL method. However, we also compare its performance with that of Split ImageNet-R [25], as this is a key benchmark proposed in DualPrompt. ImageNet-R [9] comprises of a collection of images with diverse styles for 200 classes out of the 1,000 classes in ImageNet. In experiments, DAP shows a notable improvement over L2P and DualPrompt (Row (A) of Table 8). We summarize the detailed specifications for Split ImageNet-R in Supp. A. Experimentally, the utilization of SGD yielded favorable outcomes in terms of learning accuracy (Lrn Acc) on Split ImageNet-R. Excluding prompt insertion in the final (12th) layer, rather than inserting prompts in all layers, also helped enhance stability in performance.

To address the possible concern that DAP might be only effective on benchmarks with a small number of classes and short task sequences, we compare the performance on Split DomainNet [19] which consists of 345 classes split into 15 tasks (23 classes) or 69 tasks (5 classes). As shown in rows (B) and (C) of Table 8, DAP maintains its superiority on Split DomainNet. DAP consistently outperforms on more complex benchmarks with a longer horizon.

H. Structural Similarity to Hypernetworks

The proposed adaptive prompt generator exhibits certain structural similarities with hypernetworks [6], also known as weight generators that are conditioned on an input instance or a task embedding. Similar to our prompt generator, hypernetworks also generate learnable weights. However, while hypernetworks generate the parameters of the model, our prompt generator generates input tokens that provide instructions for effectively utilizing the pre-trained representation space.

To validate the effectiveness of our approach, we compare the performance of a representative hypernetwork-based continual learning method [22] on Split CIFAR-100. However, because HNET [22] is an architecture-based method based on model weight generation, it is not feasible to migrate the proposed method and structure to pre-trained transformer-based models. For a relatively fair comparison, we instead allow a rehearsal-based approach for HNET (with a buffer size of 1000). However, as shown in Table 9, DAP outperforms HNET by a significant margin.

Split CIFAR-100	Buffer	Avg Acc (↑)	Forgetting (↓)	Lrn Acc (↑)
HNET [22]	O	42.07±1.19	5.32±3.04	46.86±3.04
DAP	X	94.05±1.19	2.28±0.96	96.37±0.74

Table 9. Comparison results of HNET and DAP on Split CIFAR-100.

Conceptually, our prompt generator can be classified as a type of generative model. However, it is the first prompt-generation approach that utilizes a generative model to create an instance-level prompt for each input instance.

I. Limitation and Discussion

We present a novel prompt-based CL method named DAP, which successfully resolves the domain scalability problem of current prompt-based CL methods. However, implementing DAP to various pre-trained models requires additional considerations due to the inherent characteristics of each model. For example, self-supervised models being pre-trained on ImageNet with ViTs such as MAE [7] and MoCo v3 [2] may not have a feature representation space sufficient to make a correct prediction on the benchmark distant from ImageNet by only relying on instructions from the prompt [10]. However, to tackle the domain scalability problem in the same (fair) setting, we first had to follow the choice of the pre-trained model of current prompt-based CL methods. Thus, the analysis of DAP with different pre-trained models is lacking. Since the motivation of this work is to address the domain scalability of the prompt-based CL methods, we decide to delve into the role and impact of various pre-trained models on prompt-based CL as future work.

As an alternative for generating domain-adaptive prompts, a multi-head attention module can be used instead of `MLP` with input transpose to obtain the global relation of input patch tokens. However, in this case, additional ideas such as pooling or a selection module are needed because directly generating the prompts as desired length is impossible. Instead of `LT`, it is possible to train a prompt mask based on the estimated key.

J. Potential Negative Social Impact

It is relatively unlikely that the dataset and problem covered by CL become potential harms to a specific social community. Also, since the prompt-based CL methods do not require heavy GPU computation, potential environmental negative impacts will be insignificant. However, different from conventional CL benchmarks, this work also deals with the generalization performance on benchmarks of specialized domains. Our testing benchmarks contain not only the natural domain but also the aerial and medical domains. These two domains can have a relatively high ethical standard. By noting it, we utilize commonly used public data of aerial and medical domains. In the case of ISIC and ChestX, belonging to the medical domain, personal identity is hidden and cannot be specified.

References

[1] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018. 4

[2] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 7

[3] 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. 2

[4] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. 2

[5] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 2019. 5

[6] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 6

[7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 7

[8] 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. 2

[9] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021. 2, 6

[10] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 7

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 3

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 3

[14] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *CVPR*, pages 6979–6987, 2017. 4

[15] Magdalena Main-Knorn, Bringfried Pflug, Jerome Louis, Vincent Debaecker, Uwe Müller-Wilm, and Ferran Gascon. Sen2cor for sentinel-2. In *Image and Signal Processing for Remote Sensing XXIII*, volume 10427, pages 37–48. SPIE, 2017. 2

[16] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016. 2

[17] Jaehoon Oh, Sungnyun Kim, Namgyu Ho, Jin-Hwa Kim, Hwanjun Song, and Se-Young Yun. Understanding cross-domain few-shot learning: An experimental study. *arXiv preprint arXiv:2202.01339*, 2022. 1

[18] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012. 2

[19] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 2, 6

[20] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2018. 5

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[22] Johannes Von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019. 7

[23] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, pages 2097–2106, 2017. 2

[24] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *CVPR*, pages 834–843, 2021. 5

[25] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. DualPrompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022. 3, 6

[26] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pages 139–149, 2022. 3