Detrimental Memories in Transfer Learning

Amal Alnouri [1] **Timothy J Wroge** [2] **Bilal Alsallakh** [2]

Abstract

The source domain in transfer learning provides essential features that enable effective and data-efficient learning on the target task. Typically, the finetuning process does not explicitly account for how the knowledge about the source domain interacts with the target task. We demonstrate how that knowledge can interfere with the target task leading to negative transfer. Specifically, certain memories about the source domain can distract the finetuned model in certain inputs. We provide a method to analyze those memories in typical foundational models and to surface potential failure cases of those models. This analysis helps model developers explore remedies for those failure cases. Our results can be reproduced at `https://github.com/AmAlnouri-JKU/TL_Interference`

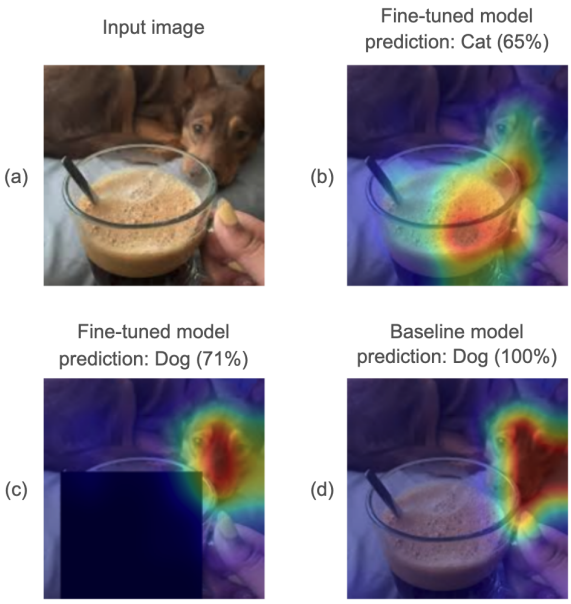

Figure 1. Cats-vs-dogs classification of a challenging input (a), using: (b, c) A model pretrained on ImageNet vs. (d) the same model trained from scratch. The fine-tuned model recognizes the `espresso` instance, which interferes with its prediction, as the GradCAM heatmap reveals (b). Masking this instance (c) or training from scratch (d) mitigates the interference.

1. Introduction

Consider a typical application of transfer learning (TL) in image classification. The task is to train a deep neural network to classify images of cats and dogs. We use the Dogs-vs-Cats dataset (Cukierski, 2013) to finetune a ResNet-18 model (He et al., 2016), pretrained on ImageNet (Deng et al., 2009). The model reaches 97.8% validation accuracy.

What happens when the input contains instances of ImageNet classes besides cats and dogs? Figure 1 demonstrates the prediction results for a dog image that features a visually prominent instance of the `espresso` class. As evident in Figure 1b, the model classifies the input as `cat`, mostly based on the region occupied by the coffee mug as evident in the GradCAM heatmap (Selvaraju et al., 2017). Interestingly, when this region is occluded, the model correctly classifies the input, focusing on salient features of the `Dog` class (Figure 1c). These observations suggest that the finetuned model is still able to recognize certain ImageNet classes. Moreover, this ability can interfere with the target task. Training the model from scratch seems to mitigate this interference (Figure 1d).

We present a method to expose source-domain knowledge that can interfere with the target task in TL, focusing on image classification and on finetuning pretrained weights as a dominant form of TL. Our contributions are:

- Investigating how source memories can interfere with the target task (Section 2).
- Proposing methods to expose this interference (Section 3).
- Demonstrating how the above-mentioned interfering memories explain real-world failure cases (Section 4).

We elaborate on related work in Section 5 and discuss potential solutions to mitigate the interference in future work.

[1]Damascus University [2]Voxel AI, San Francisco, United States. Correspondence to: Bilal Alsallakh <bilal@voxelai.com>.

Work presented at TF2M workshop at ICML 2024, Vienna, Austria. PMLR 235, 2024.

2. Our Hypothesis: Lingering Memories

We refer to the dataset a model is pretrained on as the source domain, and to the dataset this model is fine-tuned on as the target domain. Also, we refer to the process of training the model on the target task with the pretrained weights used for initialization as the standard fine-tuning (SFT) paradigm for transfer learning. This process might involve adding new randomly-initialized layers such as a linear classification head. The process might also involve freezing the weights of certain pretrained layers.

Our hypothesis for the misprediction we observe in Section 1 is that the standard finetuning (SFT) paradigm for TL is inherently inadequate: This paradigm does not explicitly account for how the source knowledge interacts with the target task. Specifically, input features that manifest frequently in the source domain and rarely in the target domain can be challenging and potentially detrimental.

For example, consider a source domain where instances of `espresso` are abundant. A model pretrained on this domain is likely to learn this feature, especially if it is relevant for the pretraining task. Now consider a target domain that contains no visual instances of `espresso` in its training images. In the SFT paradigm, such irrelevant memories might become partially or completely lost in case the pretrained weights are not frozen, a phenomenon called Catastrophic Forgetting. Nevertheless those memories might also be largely retained, since the SFT paradigm has no mechanism to explicitly destroy them. In that case, a test-time input that contains `espresso` can activate those memories as evident in Figure 1b, an edge case the model did not encounter during the finetuning process. Those *lingering memories* become detrimental as they make the predictions of the fine-tuned model arbitrary when activated. In contrast, trained the same model "from scratch" starting with random initialization can handle the same input correctly, as evident in (Figure 1d).

3. Exposing Lingering Memories

We call input features that manifest frequently in the source domain and rarely in the target domain as *source-only features*. Those features could represent object or scene categories, as well as low-level visual concepts, e.g. zig-zag patterns (Kim et al., 2018). Lingering memories are source-only features that continue to be recognized by the finetuned model. Without loss of generality, we utilize ImageNet classes to demonstrate how we expose those memories.

3.1. Identifying source-only features

We denote by $\mathrm{M^s}$ a model pretrained on a source dataset D^s and by $\mathrm{M^t}$ the same model after finetuning on the target dataset D^t using the SFT paradigm. We denote by M^b a baseline version of the model trained "from scratch" starting with random initialization. We denote by $\mathrm{C}(D)$ the classes of D. We denote by $\mathrm{H}(\mathrm{M})$ the classification head of M and by $\mathrm{B}(\mathrm{M})$ the remaining layers in M, often called the backbone. We leverage the predictions of $\mathrm{M^s}$ to assess the visual content of an image in D^t. A class among the top-k predictions suggests that the image contains visual features related or similar to it. We compute a score for each class $c_s \in \mathrm{C}(D^s)$ to quantify the prevalence of those features in D^t:

$$\mathrm{v}(c_s) = \left|x \in D^t : c_s \in \mathrm{top_k}(\mathrm{M^s(x)})\right| \tag{1}$$

A relatively high value of $v(c_s)$ indicates that the corresponding visual features manifest frequently in D^t. Figure 4 provides examples of ImageNet classes that do or do not manifest in the Cats-vs-Dogs dataset. Among $\mathrm{C}(D^s)$ we identify source-only features as:

$$F^{s\setminus t} = \left\{c \in \mathrm{C}(D^s) : \mathrm{v}(c) \leq v^{\mathrm{lo}}\right\} \tag{2}$$

where k and v^{lo} are parameters we choose depending on the source and target datasets.

3.2. Identifying lingering features

A source-only feature c is lingering if the target model can recognize it in a given input. To quantify this ability, we construct a new model that applies the fine-tuned backbone layers $\mathrm{B}(\mathrm{M^t})$ followed by the source classification head $\mathrm{H}(\mathrm{M^s})$. The new model, denoted by $\mathrm{M^{st}}$, can recognize a source-only feature if $\mathrm{B}(\mathrm{M^t})$ has sufficient signal about it:

$$L = \{c \in F^{s\setminus t} : \mathrm{mean}(\{p_c(\mathrm{M^{st}}(x)) : x \in D^c\}) > p^{\mathrm{lo}}\} \tag{3}$$

where p_c denotes the prediction score of class c, D^c is the subset of D^s labeled as c, and p^{lo} is a threshold we choose based on the source model characteristics.

As a qualitative evidence of lingering features, consider the input images in Figure 2. Both the `Espresso` class and the `Cock` class are source-only features as illustrated in Figure 4. The finetuned model can still recognize instances of those classes as the corresponding GradCAM heatmaps suggest, both when present alone (first row) and besides a target class (third row). In contrast, when the model is trained from scratch, the heatmaps focus only on the target class as evident in the third row. Moreover, the heatmaps for this model are arbitrary in the first row, where the target classes do not manifest in the input. This is expected since this model is indifferent to source-only features.

4. Impact of Lingering Memories

Lingering memories might impact the target task if the corresponding features manifest in the input. However, since they correspond to source-only features, these memories

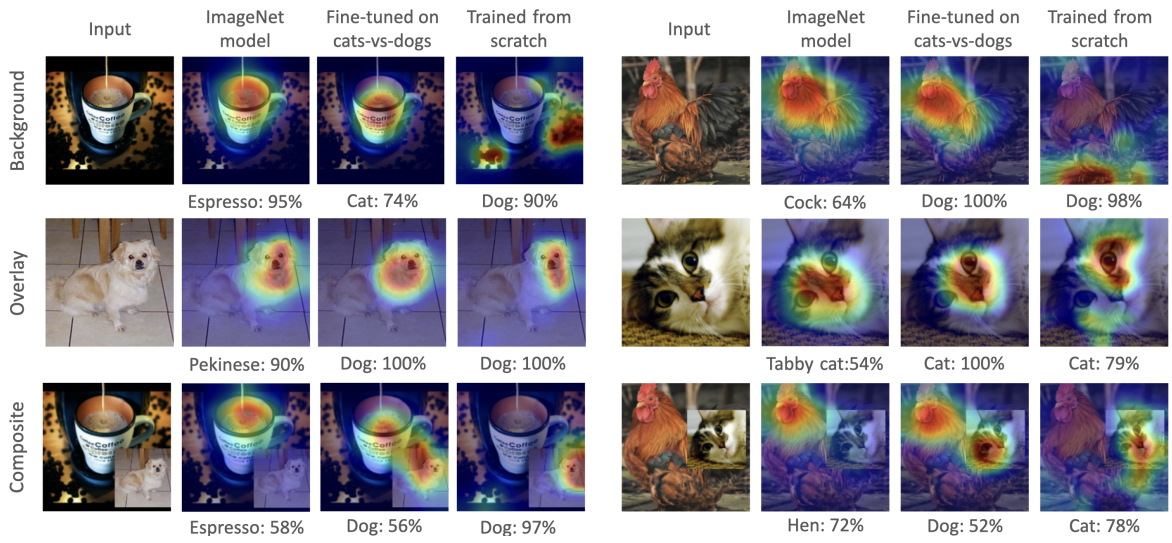

Figure 2. Demonstrating lingering memories using Grad-CAM attribution maps. The first row depicts how two ImageNet samples are processed by the source ResNet-18 model $\mathrm{M^s}$, fine-tuned model $\mathrm{M^t}$, and baseline model $\mathrm{M^b}$. The samples are of source-only classes whose features are still recognizable by $\mathrm{M^t}$ as evident in the heat maps. The second row illustrates two samples from the target dataset. The third row illustrates composite inputs where the target samples (2nd row) are randomly overlaid over the ImageNet samples (1st row).

remain dormant when feeding M^t with data points from D^t. We synthesize inputs that activate these memories in order to systematically analyze how they impact M^t. We further demonstrate this impact using natural test images we crawled beyond D^t.

For each class in L we construct composite images that contain both an instance of c_s and an instance of a target class c_t. For this purpose, we superimpose a random image from D^t as an overlay at a random location on top of each instance of c_s in the validation subset of D^s. We denote the resulting set of composite images by $I(c_s)$. The overlay is downscaled by 40% along both dimensions, covering only $0.4^2 = 16\%$ of the image area and leaving significant details about c_s as demonstrated in the third row of Figure 2. To identify which pixels lead M^t to predict a target label c_t for a given input $x \in I(c_s)$, we compute a class activation map $A(x, M^t, c_t)$ using GradCAM. This map represents the importance of each pixel to the model prediction.

A distraction has likely happened if $A(x, M^t, c_t)$ and $A(x, M^s, c_s)$ resemble each other outside the overlay region. To quantify this resemblance, we decompose each attribution map into two regions $A = A^o + A^b$ that correspond to the overlay region and to the background region respectively. We further focus on the pixels whose heatmap values are above the 80% percentile, denoted by $\mathrm{P}_{80}(A)$, and compute the intersection over union between them:

$$\mathrm{distraction}(x, M) = \mathrm{IoU}(\mathrm{P}_{80}(A^b(x, M, c_t)), \mathrm{P}_{80}(A^b(x, M^s, c_s)) \quad (4)$$

Figure 3a plots $\mathrm{distraction}(x, M^t)$ vs. $\mathrm{distraction}(x, M^b)$ for all classes $c_s \in L$, averaging over the respective samples $x \in I(c_s)$. Both M^t and M^b are ResNet-18 models trained on the Cats-vs-Dogs dataset. We computed L with $p^{\mathrm{lo}} = 0.01$ which is 10 times higher than the expected score with a no-skill ImageNet classifier. The fine-tuned model is significantly more likely to be distracted than the baseline model in the presence of lingering features.

Impact on real-world images We crawled a variety of real images that contain cats or dogs besides objects that correspond to lingering features. Figure 3b demonstrates how the fine-tuned ResNet-18 can be distracted in the presence of such features, leading to wrong predictions. Figures 7- 8 provide a variety of additional examples with various architectures trained under different schemes (supervised and self-supervised). It further demonstrates how occluding the distracting objects results in accurate predictions and GradCAM heatmaps (Figure 5). In contrast the fine-tuned model is not distracted in the presence of source features that are common in the target dataset, as evident in Figure 3c (further examples in Figure 6).

5. Discussion

Negative Transfer (NT) is a well-known phenomenon in TL (Pan & Yang, 2009; Weiss et al., 2016), where knowledge transfer has a negative impact on the target task. A variety of studies and techniques have been proposed to characterize and to reduce this impact. A recent survey (Zhang et al., 2022) categorizes these techniques into secure trans-

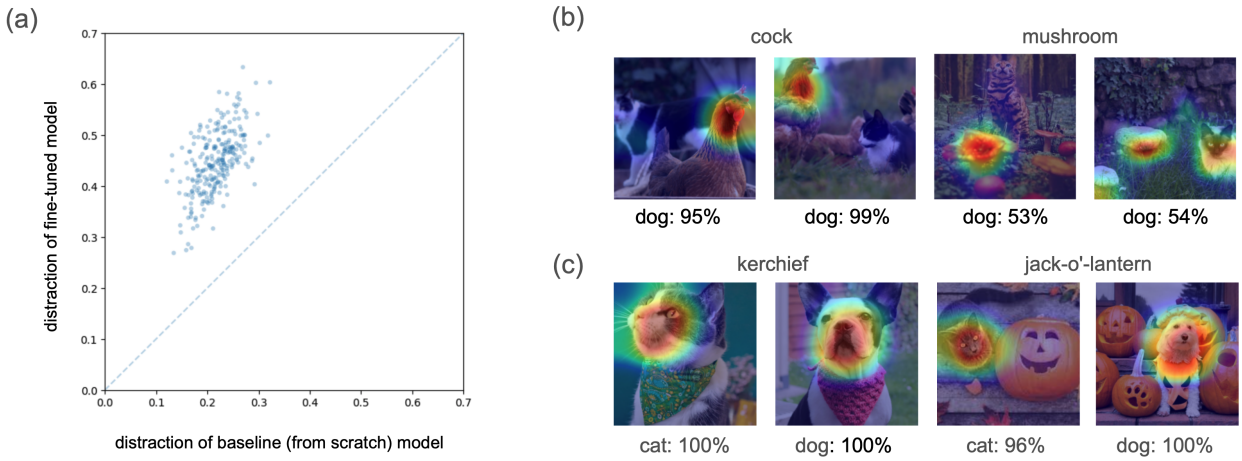

Figure 3. Impact of lingering memories. (a) Comparing the distraction of the fine-tuned and baseline models . Each dot represents a source class c_s and encodes the average distraction over 100 composite images in $I(c_s)$. (b, c) GradCAM heatmaps for real-world images that contain an instance of a target class along with (b) a source-only feature and (c) a source feature that is common in the target dataset.

fer, domain similarity estimation, distant transfer, and NT mitigation. A major root cause behind NT is the distribution differences between the source and the target domains (Seah et al., 2013; Ge et al., 2014; Wang et al., 2019).

Our work aims to provide a nuanced understanding of NT, proposing lingering memories as the mechanism in which it manifests. Our approach is inspired by the paired-associates learning paradigm in behavioural psychology (Postman & Stark, 1969), which shows how NT induces interference similar to the distractions we demonstrate. We draw further inspiration from domain adaptation theory which suggests that "for effective domain transfer to be achieved, predictions must be made based on features that cannot discriminate between the training (source) and test (target) domains" (Ganin et al., 2016).

5.1. Potential Mitigation Approaches

We outline possible NT mitigation approaches based on our characterization of lingering memories as the root cause.

Interference Detection It is possible to detect if lingering memories are present in an inference-time input x, before feeding it into he target model $\mathrm{M^t}$. We can leverage the source model $\mathrm{M^s}$ for this purpose by feeding x into this model and determining if this model can detect one of the classes that correspond to lingering memories L as follows:

$$\text{Interference} \iff p(\mathrm{M^{st}}(x)) > p^{\mathrm{lo}} \tag{5}$$

If a potential interference is detected, the output $\mathrm{M^t}(x)$ might be impacted by negative transfer. Accordingly, We can flag x as an input that should be processed in alternative ways, e.g., using a model trained from scratch or with help of human experts if the use case allows human intervention in the decision-making process.

Retraining the source model One simple mitigation of NT incurred by source-only features is to remove the corresponding classes from D^s and to transfer from a source model M^s pretrained only on the remaining classes. In fact various studies observed improved TL by pretraining on only selected subsets of ImageNet (Kucer & Oyen, 2021; Wang et al., 2019). This approach, however, defeats the purpose of Foundational Models as general-purpose pretrained backbones, that are often prohibitive to train from scratch.

Interference-aware finetuning Ultimately, we need to make the finetuning process aware of the distribution differences between D^s and D^t. Our future work aims to investigate training objectives that can unlearn source-only features during finetuning as outlined in Appendix B.

6. Conclusion

We investigated how source-domain knowledge can be sometimes detrimental to the target task in TL, focusing on finetuning of pretrained weights as the predominant paradigm of TL. We identified source-only features as a potential culprit since they almost never manifest in the target dataset. This deprives the finetuned model of the ability to learn how to properly handle inputs in which these features manifest, with respect to the target task. We demonstrated how the model can be distracted by those features, leading to erroneous inference. Our future work aims to understand which aspects of finetuning impact NT as exemplified in Figure 7 and to explore possible mitigation thereof.

References

Cukierski, W. Dogs vs. cats, 2013. URL https://kaggle.com/competitions/dogs-vs-cats.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE, 2009.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Ge, L., Gao, J., Ngo, H., Li, K., and Zhang, A. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):254–271, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Kucer, M. and Oyen, D. Transfer learning with fewer imagenet classes. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359, 2009.

Postman, L. and Stark, K. Role of response availability in transfer and interference. *Journal of Experimental Psychology*, 79(1p1):168, 1969.

Seah, C.-W., Ong, Y.-S., and Tsang, I. W. Combating negative transfer from predictive distribution differences. *IEEE Transactions on Cybernetics*, 43(4):1153–1165, 2013. doi: 10.1109/TSMCB.2012.2225102.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Wang, Z., Dai, Z., Poczos, B., and Carbonell, J. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

Zhang, W., Deng, L., Zhang, L., and Wu, D. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

A. Additional Figures and Examples

In Figure 4 we provide examples of source-only features we identified in the ImageNet dataset with respect to the Cats-vs-Dogs target dataset, in addition to source features that are common in the target dataset. For simplicity, we consider the ImageNet classes to be the source features. Nevertheless, lower-level features such as zig-zag patterns and vegetation could be considered when comparing the source and target domain.

Figure 4. Illustrating the difference between source-only features and ones shared between the source and target domains: (a) Examples of ImageNet classes that do not manifest in the target Cats-vs-Dogs dataset. (b) Examples of ImageNet classes that manifest in the Cats-vs-Dogs dataset according to Eq 1. This includes different feline and canine classes, classes that have similar visual features such as `weasel`, and classes of objects that commonly appear together with cats and dogs, as illustrated in (c). (c) Samples from the Cats-vs-Dogs datasets where `kerchief`, `jack-o'-lantern`, and `tennis ball` manifest.

In Figure 5 we demonstrate how occluding source-only features in the input eliminates the distraction of the fine-tuned model and results in accurate prediction. This provides an evidence that lingering memories about those source-only features is the root cause of distraction.

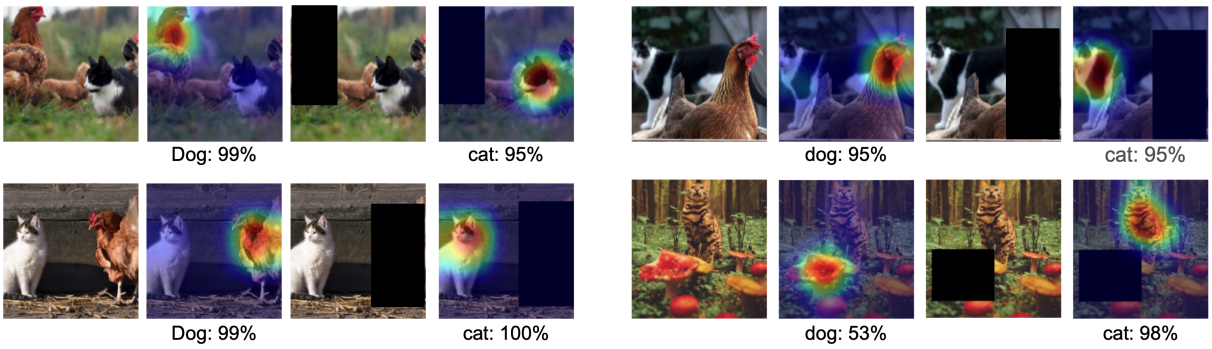

Figure 5. Prediction results and GradCAM heatmaps of the fine-tuned model for four real-world images that contain cats alongside instances of source-only classes. Occluding these instances eliminates the distraction evident otherwise.

In Figure 6 we demonstrate how features shared between the source and target domains do not confuse the fine-tuned target model.

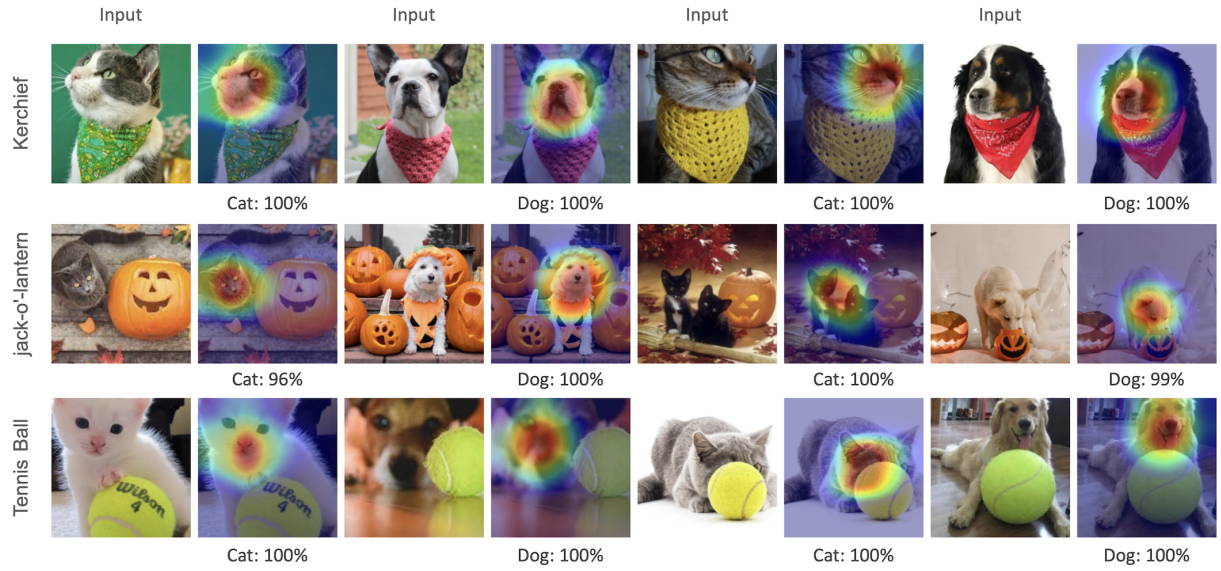

Figure 6. Prediction results and GradCAM heatmaps of the fine-tuned model for real-world images that contain the target object alongside instances of source classes that are common in the target set. Each class demonstrates examples for one source class. Notice how the fine-tuned model is not distracted by those instances, as evident in the heatmaps and the highly-accurate prediction results.

Impact of Training Set Size In Figure 7 we demonstrate how increasing the size of the target training set helps mitigate negative transfer induced by lingering memories. A large training set is more likely to destroy those memories compared with a small dataset. This is evident in the GradCAM heatmaps where the model focuses increasingly less on the distracting objects.

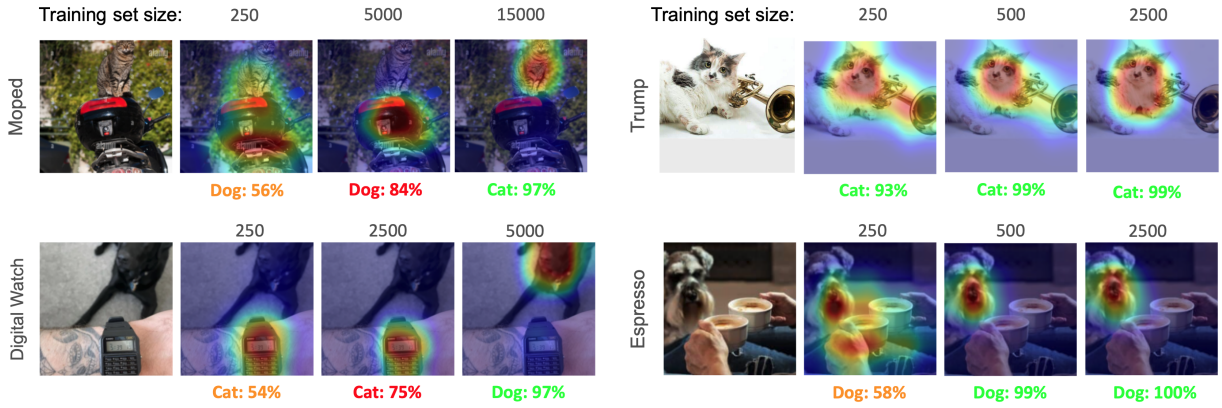

Figure 7. Demonstrating how the size of the target training set impacts the distraction of fine-tuned model. The four examples depicted contain a target class along with a source-only features. Training on a large dataset reduces the impact of those features on the fine-tuned model as evident through the prediction scores and the GradCAM heatmaps.

Impact of Model Architecture In Figure 8 we demonstrate how source-only features can trigger different lingering memories in different architectures. This depends on which source knowledge is retained or destroyed during the fine-tuning process, which can vary between different architectures. This also depends on the training hyperparameters such as the aforementioned training set size, number of iterations, and regularization techniques.

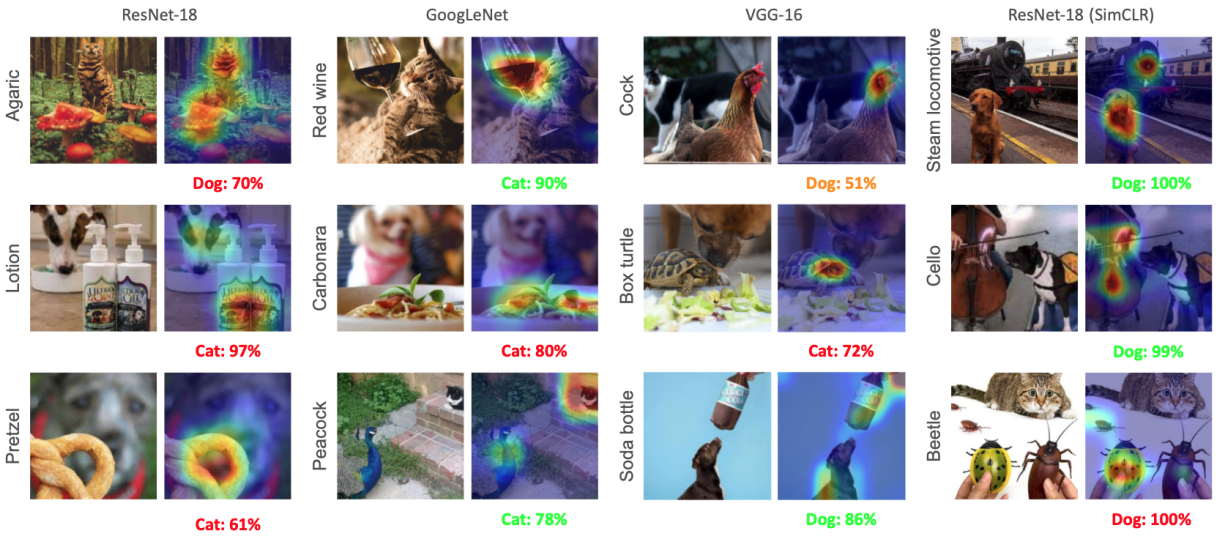

Figure 8. Demonstrating how models of various architectures can be impacted by lingering memories. Notice how different models might vary in which source-only features might be retained during training and hence might result in lingering memories and potential distraction.

B. Potential Future Work and Solutions to Negative Transfer

In this section we explore mitigation measures that may be relevant to address the issues outlined in negative transfer in this paper. There is an excellent summary of work for these solutions outlined by (Zhang et al., 2022) as well as by (Weiss et al., 2016). We would like to outline mitigation measures and other ways of approaching the problem of negative transfer. Because negative transfer is fundamentally a problem of source features contributing poorly to the target dataset, we can try to understand ways that these source features can be suppressed in the course of training.

Some straightforward solutions could solve this using regularization. If we consider the set of features for a class in the source domain (e.g. Zebra) and another class in the target domain (e.g. Cat), we can represent these features as sets:

$$F^{Zebra} = \{f_i \in F | activation(f_i, x) > a_t, \forall x \in D^{zebra}\} \tag{6}$$

$$F^{Cat} = \{f_i \in F | activation(f_i, x) > a_t, \forall x \in D^{cat}\} \tag{7}$$

For some activation threshold a_t and the total set of potential neural features as F. Given this representation, we would want to suppress any feature $\{f \in F^{zebra} | f \notin F^{cat}\}$ as these are the memories that may trigger cases of negative transfer.

Some potential mitigation measures for this may be to implement an L1 regularization to the optimization such that the network is as parsimonious as possible with its activations; this should lead to sparsity in the output of the network and decrease the chance that irrelevant features propagate and cause an unwanted affect to the predictions in the target domain.

Equivalently options like network pruning may also help in these situations. Since network pruning is primarily focused on minimizing feature activation, it may be a good candidate to mitigate cases of negative transfer.