# A Data-Based Perspective on Transfer Learning

Saachi Jain*
MIT
saachij@mit.edu

Hadi Salman*
MIT
hady@mit.edu

Alaa Khaddaj *
MIT
alaakh@mit.edu

Eric Wong
University of Pennsylvania
exwong@seas.upenn.edu

Sung Min Park
MIT
sp765@mit.edu

Aleksander Mądry
MIT
madry@mit.edu

## Abstract

*It is commonly believed that in transfer learning including more pre-training data translates into better performance. However, recent evidence suggests that* removing *data from the source dataset can actually help too. In this work, we take a closer look at the role of the source dataset's composition in transfer learning and present a framework for probing its impact on downstream performance. Our framework gives rise to new capabilities such as pinpointing transfer learning brittleness as well as detecting pathologies such as data-leakage and the presence of misleading examples in the source dataset. In particular, we demonstrate that removing detrimental datapoints identified by our framework indeed improves transfer learning performance from ImageNet on a variety of target tasks.* [1]

## 1. Introduction

Transfer learning enables us to adapt a model trained on a *source* dataset to perform better on a downstream *target* task. This technique is employed in a range of machine learning applications including radiology [23, 45], autonomous driving [11, 24], and satellite imagery analysis [44, 47]. Despite its successes, however, it is still not clear what the drivers of performance gains brought by transfer learning actually are.

So far, a dominant approach to studying these drivers focused on the role of the source *model*—i.e., the model trained on the source dataset. The corresponding works involve investigating the source model's architecture [23], accuracy [27], adversarial vulnerability [42, 43], and training procedure [21, 30]. This line of work makes it clear that the properties of the source model has a significant impact on transfer learning. There is some evidence, however, that the source *dataset* might play an important role as well [18, 26, 38]. For example, several works have shown that while increasing the size of the source dataset generally boosts transfer learning performance, *removing* specific classes can help too [18, 26, 38]. All of this motivates a natural question:

*How can we pinpoint the exact impact of the source dataset in transfer learning?*

**Our Contributions.** In this paper, we present a framework for measuring and analyzing the impact of the source dataset's composition on transfer learning performance. To do this, our framework provides us with the ability to investigate the counterfactual impact on downstream predictions of including or excluding datapoints from the source dataset, drawing inspiration from classical supervised learning techniques such as influence functions [7, 13, 25] and datamodels [19]. Using our framework, we can:

- Pinpoint what parts of the source dataset are most utilized by the downstream task.
- Automatically extract granular subpopulations in the target dataset through projection of the fine-grained labels of the source dataset.
- Surface pathologies such as source-target data leakage and mislabelled source datapoints.

We also demonstrate how our framework can be used to find detrimental subsets of ImageNet [9] that, when removed, give rise to better downstream performance on a variety of image classification tasks.

*Equal contribution.
[1] Code is available at https://github.com/MadryLab/data-transfer

## 2. A Data-Based Framework for Studying Transfer Learning

In order to pinpoint the role of the source dataset in transfer learning, we need to understand how the composition of that source dataset impacts the downstream model's performance. To do so, we draw inspiration from supervised machine learning approaches that study the impact of the training data on the model's subsequent predictions. In particular, these approaches capture this impact via studying (and approximating) the counterfactual effect of excluding certain training datapoints. This paradigm underlies a number of techniques, from influence functions [7, 13, 25], to datamodels [19], to data Shapley values [14, 22, 31].

Now, to adapt this paradigm to our setting, we study the counterfactual effect of excluding datapoints from the *source* dataset on the downstream, *target* task predictions. In our framework, we will focus on the inclusion or exclusion of entire *classes* in the source dataset, as opposed to individual examples[2]. This is motivated by the fact that, intuitively, we expect these classes to be the ones that embody whole concepts and thus drive the formation of (transferred) features. We therefore anticipate the removal of entire classes to have a more measurable impact on the representation learned by the source model (and consequently on the downstream model's predictions).

Once we have chosen to focus on removal of entire source classes, we can design counterfactual experiments to estimate their influences. A natural approach here, the *leave-one-out* method [7, 25], would involve removing each individual class from the source dataset separately and then measuring the change in the downstream model's predictions. However, in the transfer learning setting, we suspect that removing a single class from the source dataset won't significantly change the downstream model's performance. Thus, leave-one-out methodology may be able to capture meaningful influences only in rare cases. This is especially so as many common source datasets contain highly redundant classes. For example, ImageNet contains over 100 dog-breed classes. The removal of a single dog-breed class might thus have a negligible impact on transfer learning performance, but the removal of all of the dog classes might significantly change the features learned by the downstream model. For these reasons, we adapt the *subsampling* [13, 19] approach, which revolves around removing a random collection of source classes at once.

**Computing transfer influences.** In the light of the above, our methodology for computing the influence of source classes on transfer learning performance involves training a large number of models with random subsets of the source classes removed, and fine-tuning these models on the target task. We then estimate the influence value of a source class $\mathcal{C}$ on a target example $t$ as the expected difference in the transfer model's performance on example $t$ when class $\mathcal{C}$ was either included in or excluded from the source dataset:

$$\text{Infl}[\mathcal{C} \to t] = \mathbb{E}_S\left[f(t;S) \mid \mathcal{C} \subset S\right] - \mathbb{E}_S\left[f(t;S) \mid \mathcal{C} \not\subset S\right], \tag{1}$$

where $f(t;S)$ is the softmax output[3] of a model trained on a subset $S$ of the source dataset. A positive influence value indicates that including the source class $\mathcal{C}$ helps the model predict the target example $t$ correctly. On the other hand, a negative influence value suggests that the source class $\mathcal{C}$ actually hurts the model's performance on the target example $t$. We outline the overall procedure in Algorithm 1, and defer a detailed description of our approach to Appendix A.

**Algorithm 1** Estimation of source dataset class influences on transfer learning performance.

**Require:** Source dataset $\mathcal{S} = \cup_{k=1}^{K} \mathcal{C}_k$ (with $K$ classes), a target dataset $\mathcal{T} = (t_1, t_2, \cdots, t_n)$, training algorithm $\mathcal{A}$, subset ratio $\alpha$, and number of models $m$

Sample $m$ random subsets $S_1, S_2, \cdots, S_m \subset \mathcal{S}$ of size $\alpha \cdot |\mathcal{S}|$:
**for** $i \in 1$ to $m$ **do**
  Train model $f_i$ by running algorithm $\mathcal{A}$ on $S_i$
**end for**
**for** $k \in 1$ to $K$ **do**
  **for** $j \in 1$ to $n$ **do**
    $\text{Infl}[\mathcal{C}_k \to t_j] = \frac{\sum_{i=1}^{m} f_i(t_j;S_i)\mathbb{1}_{\mathcal{C}_k \subset S_i}}{\sum_{i=1}^{m} \mathbb{1}_{\mathcal{C}_k \subset S_i}} - \frac{\sum_{i=1}^{m} f_i(t_j;S_i)\mathbb{1}_{\mathcal{C}_k \not\subset S_i}}{\sum_{i=1}^{m} \mathbb{1}_{\mathcal{C}_k \not\subset S_i}}$
  **end for**
**end for**
**return** $\text{Infl}[\mathcal{C}_k \to t_j]$, for all $j \in [n], k \in [K]$

**A note on computational costs.** In order to compute transfer influences, we need to train a large number of source models, each on a fraction of the source dataset. Specifically, we pre-train 7,540 models on ImageNet, each on a randomly chosen 50% of the ImageNet dataset. This pre-training step needs to be performed only once though: these same models can then be used to fine-tune on each new target task. Overall, the whole process (training the source models and fine-tuning on target datasets) takes less than 20 days using 8 V100 GPUs[4].

Are so many models necessary? In Section A.5, we explore computing transfer influences with smaller numbers

[2] In Section 4.3, we adapt our framework to calculate more granular influences of individual source examples too.

[3] We experiment with other outputs such as logits, margins, or correctness too. We discuss the corresponding results in Appendix B.

[4] Details are in Appendix A.

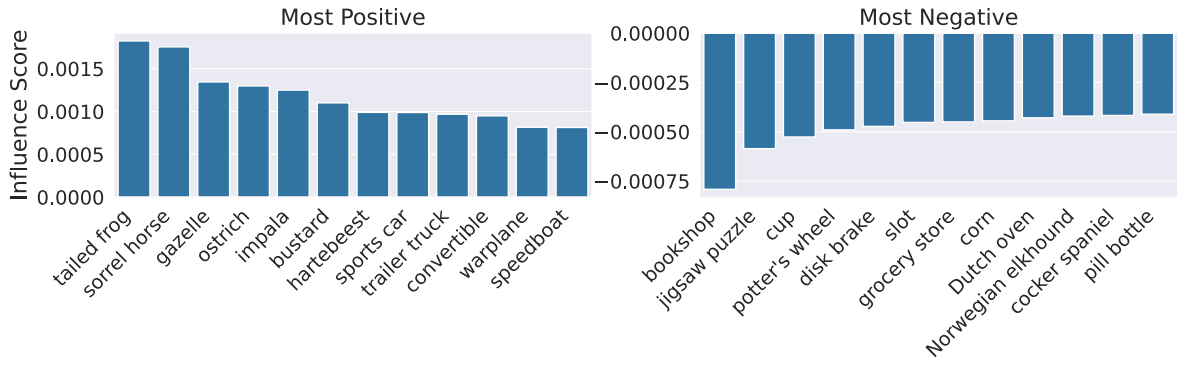
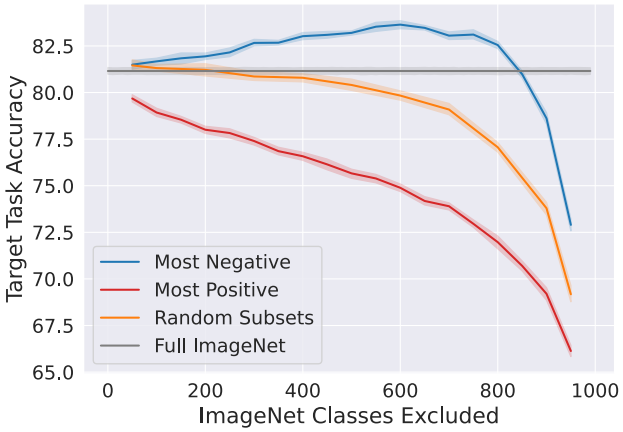



Figure 1. Most positive and negative ImageNet classes ordered based on their overall influence on the CIFAR-10 dataset. The top source classes (e.g., tailed frog and sorrel horse) turn out to be semantically relevant to the target classes (e.g., frog and horse).




(a) CIFAR-10 results

| *Target Dataset* | *Source Dataset* Full ImageNet | Removing Bottom Infl. | Semantically Relevant Classes |
|---|---|---|---|
| AIRCRAFT | $36.08 \pm 1.07$ | $\mathbf{36.88 \pm 0.74}$ | N/A |
| BIRDSNAP | $38.42 \pm 0.40$ | $\mathbf{39.19 \pm 0.38}$ | $26.74 \pm 0.31$ |
| CALTECH101 | $86.69 \pm 0.79$ | $\mathbf{87.03 \pm 0.30}$ | $82.28 \pm 0.40$ |
| CALTECH256 | $74.97 \pm 0.27$ | $\mathbf{75.24 \pm 0.21}$ | $67.42 \pm 0.39$ |
| CARS | $39.55 \pm 0.32$ | $\mathbf{40.59 \pm 0.57}$ | $21.71 \pm 0.40$ |
| CIFAR10 | $81.16 \pm 0.30$ | $\mathbf{83.64 \pm 0.40}$ | $75.53 \pm 0.42$ |
| CIFAR100 | $59.37 \pm 0.58$ | $\mathbf{61.46 \pm 0.59}$ | $55.21 \pm 0.52$ |
| FLOWERS | $\mathbf{82.92 \pm 0.52}$ | $82.89 \pm 0.48$ | N/A |
| FOOD | $56.19 \pm 0.14$ | $\mathbf{56.85 \pm 0.27}$ | $39.36 \pm 0.39$ |
| PETS | $83.41 \pm 0.55$ | $\mathbf{87.59 \pm 0.24}$ | $87.16 \pm 0.24$ |
| SUN397 | $50.15 \pm 0.23$ | $\mathbf{51.34 \pm 0.29}$ | N/A |

(b) Summary of 11 target tasks

Figure 2. Target task accuracies after removing the K most positively or negatively influential ImageNet classes from the source dataset. Mean/std are reported over 10 runs. **(a)** Results with CIFAR-10 as the target task after removing different numbers of classes from the source dataset. We also include baselines of using the full ImageNet dataset and removing random classes. One can note that, by removing negatively influential source classes, we can obtain a test accuracy that is 2.5% larger than what using the entire ImageNet dataset would yield. Results for other target tasks can be found in Appendix C. **(b)** Peak performances when removing the most negatively influential source classes across a range of other target tasks. We also compare against using the full ImageNet dataset or a subset of source classes that are semantically relevant to the target classes (defined via the WordNet hierarchy, see Appendix A for details).

of models. While using the full number of models provides the best results, training a much smaller number of models (e.g., 1000 models, taking slightly over 2.5 days on 8 V100 GPUs) still provides meaningful transfer influences. Thus in practice, one can choose the number of source models based on noise tolerance and computational budget. Further convergence results can be found in Appendix A.5.

## 3. Identifying the Most Influential Classes of the Source Dataset

In Section 2, we presented a framework for pinpointing the role of the source dataset in transfer learning by estimating the influence of each source class on the target model's predictions. Using these influences, we can now take a look at the classes from the source dataset that have the largest positive or negative impact on the overall transfer learning performance. We focus our analysis on the fixed-weights transfer learning setting (further results, including full model fine-tuning as well as generalization to other architectures, can be found in Appendix E).

As one might expect, not all source classes have large influences. Figure 1 displays the most influential classes of ImageNet with CIFAR-10 as the target task. Notably, the most positively influential source classes turn out to be directly related to classes in the target task (e.g., the ImageNet label "tailed frog" is an instance of the CIFAR class "frog"). This trend holds across all of the target datasets and transfer

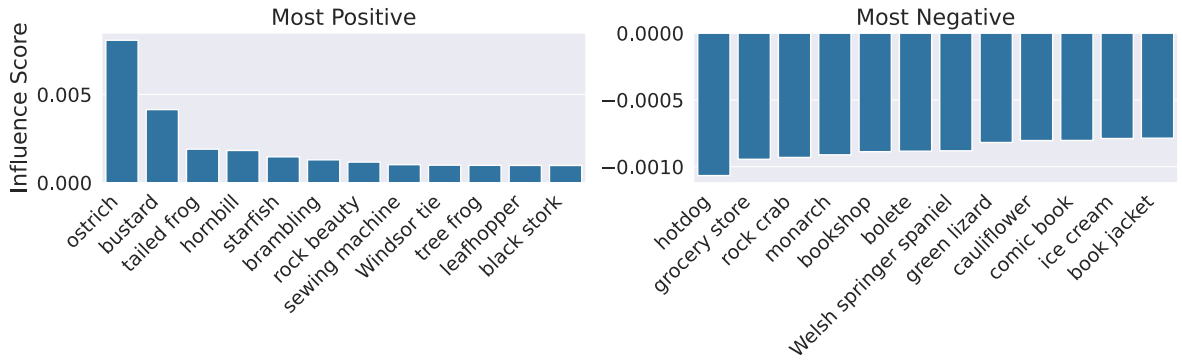



Figure 3. Most positive and negative influencing ImageNet classes for the CIFAR-10 class "bird". These are calculated by averaging the influence of each source class over all bird examples. We find that the most positively influencing ImageNet classes (e.g., "ostrich" and "bustard") are related to the CIFAR-10 class "bird". See Appendix E for results on other CIFAR-10 classes.

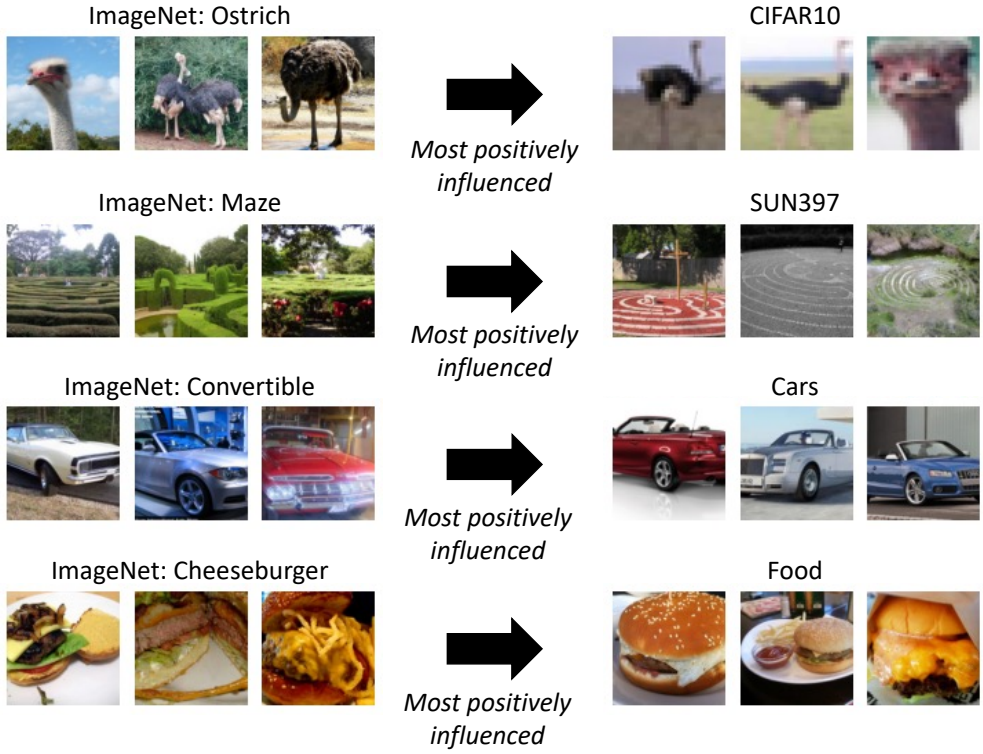



Figure 4. **Projecting source labels onto the target dataset.** For various target datasets (right), we display the images that were most positively influenced by various ImageNet classes in the source dataset (left). We find that the identified images from the target datasets look similar to the corresponding images in the source dataset.

learning settings we considered (see Appendix C). Interestingly, the source dataset also contains classes that are overall negatively influential for the target task, e.g., "bookshop" and "jigsaw puzzle" classes. (In Section 4, we will take a closer look at the factors that can cause a source class to be negatively influential for a target prediction.)

**How important are the most influential source classes?** We now remove each of the most influential classes from the source dataset to observe their actual impact on transfer learning performance (Figure 2a). As expected, removing the most positively influential classes severely degrades transfer learning performance as compared to removing random classes. This counterfactual experiment confirms that


Figure 5. The CIFAR-10 images that were most positively (or negatively) influenced by the ImageNet classes "starfish" and "rapeseed." CIFAR-10 images that are highly influenced by the "starfish" class have similar shapes, while those influenced by "rapeseed" class have yellow-green colors.

these classes are indeed important to the performance of transfer learning. On the other hand, removing the most negatively influential classes actually improves the overall transfer learning performance *beyond what using the entire ImageNet dataset provides* (see Figure 2b).

Above, we noted that the top influential source classes are typically related to the classes in the target dataset. What happens if we only choose source classes that are semantically relevant to the classes of the target dataset? Indeed, [38] found that hand-picking such source datasets can sometimes boost transfer learning performance. For each target dataset, we select ImageNet classes that are semantically relevant to the target classes (using the WordNet hierarchy, see Appendix A). As shown in Figure 2b, choosing an optimal subset of classes via transfer influences substantially outperforms this baseline.

## 4. Probing the Impact of the Source Dataset on Transfer Learning

In Section 3, we developed a methodology for identifying source dataset classes that have the most impact on transfer learning performance. Now, we demonstrate how this methodology can be extended into a framework for probing and understanding transfer learning, including: (1) identifying granular target subpopulations that correspond to source classes, (2) debugging transfer learning failures, and (3) detecting data leakage between the source and target datasets. We focus our demonstration of these capabilities on a commonly-used transfer learning setting: ImageNet to CIFAR-10 (experimental details are in Appendix A).

### 4.1. Capability 1: Extracting target subpopulations by projecting source class labels

Imagine that we would like to find all the ostriches in the CIFAR-10 dataset. This is not an easy task as CIFAR-10 only has "bird" as a label, and thus lacks sufficiently fine-grained annotations. Luckily, however, ImageNet *does* contain an ostrich class! Our computed influences enable us to "project" this ostrich class annotation (and, more broadly, the fine-grained label hierarchy of our source dataset) to find this subpopulation of interest in the target dataset.

Indeed, our examination from Section 3 suggests that the most positively influencing source classes are typically those

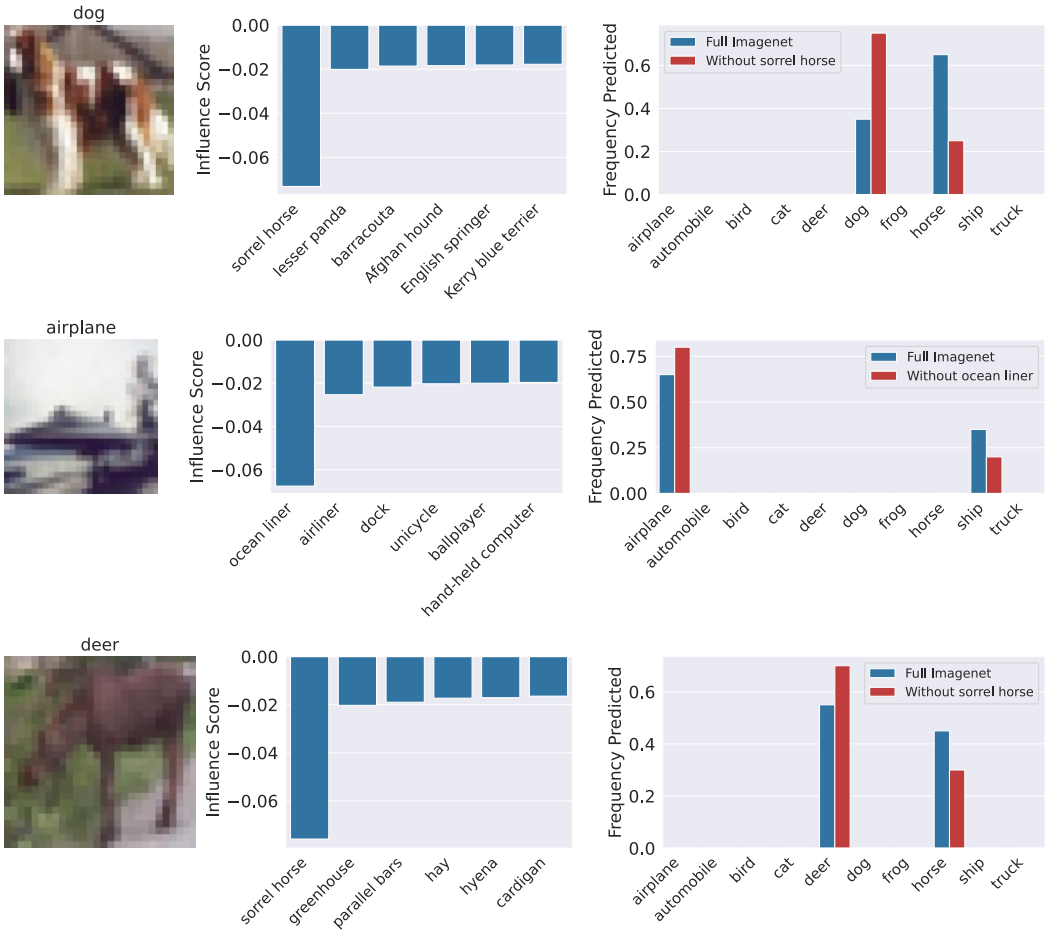



Figure 6. Pinpointing highly negatively influential source classes can help explain model mistakes. **Left:** For three CIFAR-10 images, we plot the most negatively influential source classes. **Right:** Over 20 runs, the fraction of times that our downstream model predicts each label for the given CIFAR-10 image. When the most negatively influential class is removed, the model predicts the correct label more frequently. More examples can be found in Appendix E.

that directly overlap with the target classes (see Figure 1). In particular, for our example, "ostrich" is highly positively influential for the "bird" class (see Figure 3). To find ostriches in the CIFAR-10 dataset, we thus need to simply surface the CIFAR-10 images which were most positively influenced by the "ostrich" source class (see Figure 4).

It turns out that this type of projection approach can be applied more broadly. Even when the source class is not a direct sub-type of a target class, the downstream model can still leverage salient features from this class — such as shape or color — to predict on the target dataset. For such classes, projecting source labels can extract target subpopulations which share such features. To illustrate this, in Figure 5, we display the CIFAR-10 images that are highly influenced by the classes "starfish" and "rapeseed" (both of which do not directly appear in the CIFAR-10 dataset). For these classes, the most influenced CIFAR-10 images share the same shape ("starfish") or color ("rapeseed") as their ImageNet counterparts. More examples of such projections can be found in Appendix E.

### 4.2. Capability 2: Debugging the failures of a transferred model

Our framework enables us to also reason about the possible mistakes of the transferred model caused by source dataset classes. For example, consider the CIFAR-10 image of a dog in Figure 6, which our transfer learning model often mispredicts as a horse. Using our framework, we can demonstrate that this image is strongly negatively influenced by the source class "sorrel horse." Thus, our downstream model may be misusing a feature introduced by this class. Indeed, once we remove "sorrel horse" from the source dataset, our model predicts the correct label more frequently. (See Appendix E for more examples, as well as a quantitative

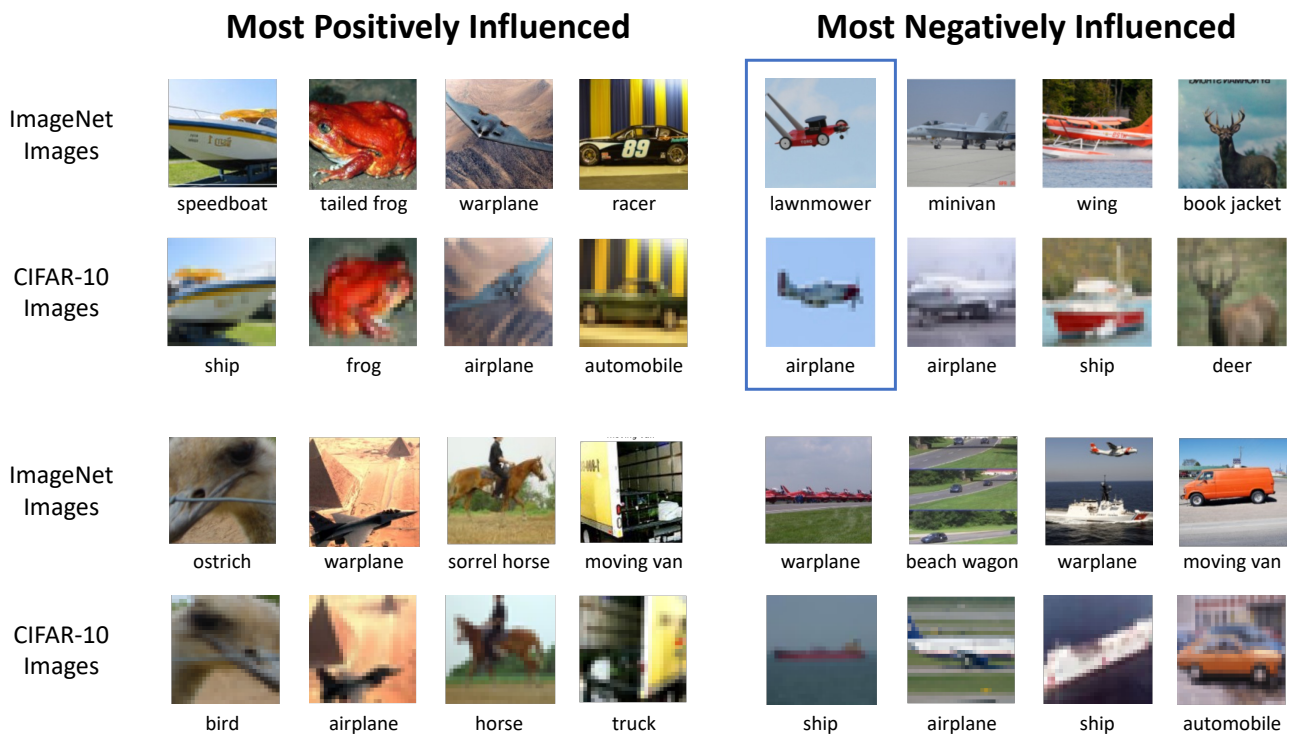



Figure 7. ImageNet training images with highest positive (**left**) or negative (**right**) example-wise (average) influences on CIFAR-10 test images. We find that ImageNet images that are highly positively influential often correspond to data leakage, while ImageNet images that are highly negatively influential are often either mislabeled, ambiguous, or otherwise misleading. For example, the presence of a flying lawnmower in the ImageNet dataset hurts the downstream performance on a similarly shaped airplane (boxed).

analysis of this experiment.)

### 4.3. Capability 3: Detecting data leakage and misleading source examples

Thus far, we have focused on how the *classes* in the source dataset influence the predictions of the transferred model on target examples. In this section, we extend our analysis to the *individual* datapoints of the source dataset. We do so by adapting our approach to measure the influence of each individual source datapoint on each target datapoint. Further details on how these influences are computed can be found in Appendix D.

Figure 7 displays the ImageNet training examples that have highly positive or negative influences on CIFAR-10 test examples. We find that the source images that are highly positively influential are often instances of *data leakage* between the source training set and the target test set. On the other hand, the ImageNet images that are highly negatively influential are typically mislabeled, misleading, or otherwise surprising. For example, the presence of the ImageNet image of a flying lawnmower hurts the performance on a CIFAR-10 image of a regular (but similarly shaped) airplane (see Figure 7).

## 5. Related Work

**Transfer learning.** Transfer learning is a technique commonly used in domains ranging from medical imaging [23, 36], language modeling [6], to object detection [5, 8, 15, 41]. Therefore, there has been considerable interest in understanding the drivers of transfer learning's success. For example, by performing transfer learning on block-shuffled images, [37] demonstrate that at least some of the benefits of transfer learning come from low-level image statistics of source data. There is also an important line of work studying transfer learning by investigating the relationship between different properties of the source model and performance on the target task [23, 27, 42, 43].

The works that are the most relevant to ours are those which studied how modifying the source dataset can affect the downstream performance. For example, [26] showed that pre-training with an enormous source dataset (approximately 300 million) of noisily labeled images can outperform pre-training with ImageNet. [1, 18] investigated the importance of the number of classes and the number of images per class in transfer learning. Finally, [38] demonstrated that more pre-training data does not always help, and transfer learning can be sensitive to the choice of pre-training data. They also

presented a framework for reweighting the source datapoints in order to boost transfer learning performance.

**Influence functions and datamodels.** Influence functions are well-studied statistical tools that have been recently applied in machine learning settings [7, 17, 25]. For a given model, influence functions analyze the effect of a training input on the model's predictions by estimating the expected change in performance when this training input is added or removed. In order to apply this tool in machine learning, [25] propose estimating the influence functions using the Hessian of the loss function. A recent line of work estimates this quantity more efficiently by training on different subsets of the training set [13]. In a similar vein, [14] proposed running a Monte Carlo search to estimate the effect of every training input via Shapley values. More recently, [19] proposed datamodeling framework as an alternative way to estimate the effect of a training input on the models' prediction. Datamodels are represented using parametric functions (typically, linear functions) that aim to map a subset of the training set to the model's output.

## 6. Conclusions

In this work, we presented a new framework for examining the impact of the source dataset in transfer learning. Specifically, our approach estimates the influence of a source class (or datapoint) that captures how including that class (or datapoint) in the source dataset impacts the downstream model's predictions. Leveraging these estimates, we demonstrate that we can improve the transfer learning performance on a range of downstream tasks by identifying and removing detrimental datapoints from the source dataset. Furthermore, our framework enables us to identify granular subpopulations in the target dataset by projecting fine-grained labels from the source dataset, better understand model failures on the downstream task and detect potential data-leakages from the source to the downstream dataset. We believe our framework provides a new perspective on transfer learning: one that enables us to perform a fine-grained analysis of the impact of the source dataset.

## References


[1] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 2015.

[2] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, 2014.

[4] Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: model-x knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[6] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. *Language Resources and Evaluation Conference (LREC)*, 2018.

[7] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems (NeurIPS)*, 2016.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[11] Shuyang Du, Haoli Guo, and Andrew Simpson. Self-driving car steering angle prediction based on image recognition. *arXiv preprint arXiv:1912.05440*, 2019.

[12] Li Fei-Fei, Rob Fergus, and Pietro 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. IEEE, 2004.

[13] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 2881–2891, 2020.

[14] Amirata Ghorbani and James Zou. Data shapley: Eq-

uitable valuation of data for machine learning. In *International Conference on Machine Learning (ICML)*, 2019.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *computer vision and pattern recognition (CVPR)*, pages 580–587, 2014.

[16] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[17] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

[18] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[19] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *International Conference on Machine Learning (ICML)*, 2022.

[20] Saachi Jain, Hadi Salman, Eric Wong, Pengchuan Zhang, Vibhav Vineet, Sai Vemprala, and Aleksander Madry. Missingness bias in model debugging. In *International Conference on Learning Representations*, 2022.

[21] Yunhun Jang, Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Learning what and where to transfer. In *International Conference on Machine Learning*, pages 3030–3039. PMLR, 2019.

[22] Bojan Karlaš, David Dao, Matteo Interlandi, Bo Li, Sebastian Schelter, Wentao Wu, and Ce Zhang. Data debugging with shapley importance over end-to-end machine learning pipelines. *arXiv preprint arXiv:2204.11131*, 2022.

[23] Alexander Ke, William Ellsworth, Oishi Banerjee, Andrew Y Ng, and Pranav Rajpurkar. Chextransfer: performance and parameter efficiency of imagenet models for chest x-ray interpretation. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 116–124, 2021.

[24] Jiman Kim and Chanjong Park. End-to-end ego lane estimation based on sequential transfer learning for self-driving cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 30–38, 2017.

[25] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017.

[26] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 2019.

[27] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *computer vision and pattern recognition (CVPR)*, 2019.

[28] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Technical report*, 2009.

[30] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[31] Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

[32] Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. ffcv. https://github.com/libffcv/ffcv/, 2022.

[33] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. 2021.

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

[35] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[36] Romain Mormont, Pierre Geurts, and Raphaël Marée. Comparison of deep transfer learning strategies for digital pathology. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[37] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[38] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

[39] 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*, 2008.

[40] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NeurIPS)*, 2015.

[42] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish

Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[43] Francisco Utrera, Evan Kravitz, N. Benjamin Erichson, Rajiv Khanna, and Michael W. Mahoney. Adversarially-trained deep nets transfer better. In *ArXiv preprint arXiv:2007.05869*, 2020.

[44] Sherrie Wang, George Azzari, and David B Lobell. Crop type mapping without field-level labels: Random forest transfer and unsupervised clustering techniques. *Remote sensing of environment*, 222:303–317, 2019.

[45] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[46] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[47] Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.