Disentangling Transfer and Interference in Multi-Domain Learning

Yipeng Zhang[1], **Tyler L. Hayes**[2], **Christopher Kanan**[234]

[1]University of Rochester, Rochester, NY, USA
[2]Rochester Institute of Technology, Rochester, NY, USA
[3]Paige, New York, NY, USA
[4]Cornell Tech, New York, NY, USA
yzh232@u.rochester.edu, {tlh6792, kanan}@rit.edu

Abstract

Humans are incredibly good at transferring knowledge from one domain to another, enabling rapid learning of new tasks. Likewise, transfer learning has enabled enormous success in many computer vision problems using pretraining. However, the benefits of transfer in multi-domain learning, where a network learns multiple tasks defined by different datasets, has not been adequately studied. Learning multiple domains could be beneficial, or these domains could interfere with each other given limited network capacity. Understanding how deep neural networks of varied capacity facilitate transfer across inputs from different distributions is a critical step towards open world learning. In this work, we decipher the conditions where interference and knowledge transfer occur in multi-domain learning. We propose new metrics disentangling interference and transfer, set up experimental protocols, and examine the roles of network capacity, task grouping, and dynamic loss weighting in reducing interference and facilitating transfer.

1 Introduction

Convolutional Neural Networks (CNNs) have achieved great success in a variety of computer vision tasks, including image classification, object detection, and semantic segmentation (Zamir et al. 2018). Although inputs for a particular task can come from various domains, many studies develop models that only solve one task on a single domain. In contrast, humans and animals learn multiple tasks at the same time and utilize task similarities to make better task-level decisions. Inspired by this phenomenon, *multi-task learning (MTL)* seeks to jointly learn a single model for various tasks, typically on the same input domain (Thrun 1996). However, in the real world, visual inputs come from several different domains, where an agent must maintain performance on all domains and facilitate transfer among inputs from similar domains. Thus, *multi-domain learning (MDL)* takes the problem a step further and requires models to learn from multiple tasks covering various domains (Bilen and Vedaldi 2017). By jointly learning feature representations, MTL and MDL models can achieve superior per-task performance than models trained on a single task in isolation. This is a result of *positive knowledge transfer* (Kendall, Gal, and Cipolla 2018). Facilitating knowledge transfer is a critical step in developing agents that can improve their performance over time on evolving data streams (Mundt et al. 2020; Parisi et al. 2019).

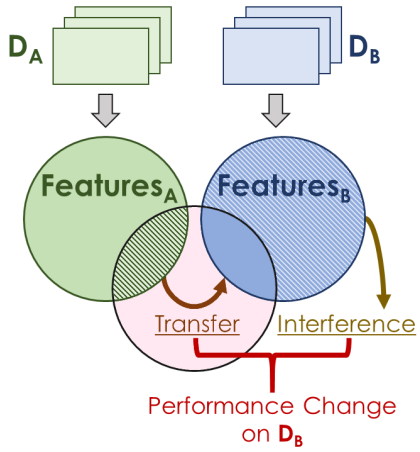


Figure 1: Given two networks trained on domains D_A and D_B respectively, domain specific features are learned. After jointly training a network on both domains, performance changes on domain D_B are influenced by transfer (shaded green) and interference (shaded blue) from D_A.

Unfortunately, jointly training models on multiple tasks does not guarantee performance gains (Zhang et al. 2020; Wu, Zhang, and Ré 2019). Past work has proposed this can occur due to sample imbalance among tasks (Wu, Zhang, and Ré 2019; Hernandez et al. 2021). We hypothesized that transfer is also affected by a lack of sufficient network capacity to learn all tasks and that task similarity can affect transfer.

In this paper, we argue that performance differences between an MDL network versus a single-domain independent network stem from the interplay between interference across tasks and the unseen knowledge transferred from other domains/tasks. We propose new metrics and establish experimental protocols to answer this question and empirically evaluate models in the MDL setting using a classification task on three common image datasets. While existing studies have focused on the contribution of task sample sizes in transfer (Wu, Zhang, and Ré 2019; Hernandez et al. 2021), we study three additional factors that could affect the amount of network transfer: 1) network capacity, 2) domain/task

groupings, and, 3) task loss weightings (Kendall, Gal, and Cipolla 2018; Groenendijk et al. 2021). In real-world scenarios, making a trade-off between capacity and performance is critical. Intuitively, a model that has more capacity could allocate more task-specific neurons. Further, more similar tasks should yield more transfer and dynamically weighing task-specific losses has been shown to be effective in regulating network learning speed and resolving optimization conflicts of different objectives in MTL (Sermanet et al. 2014; Kokkinos 2017; Eigen and Fergus 2015; Zhang et al. 2018). Thus, these loss weighting schemes could potentially reduce interference across tasks. Moreover, the effectiveness of loss weighting schemes has not been explored in the MDL setting.

This paper makes the following contributions:

1. We propose metrics for task interference and knowledge transfer calculated on a sample-wise basis, which are more comprehensive and robust compared to prior metrics.
2. We introduce an experimental framework for studying knowledge transfer empirically in the multi-domain learning regime, and comprehensively study the individual role of network capacity, task groupings, and multi-domain loss weighting schemes in facilitating knowledge transfer.

2 Related Work

2.1 Multi-Task and Multi-Domain Learning

Deep MTL architectures can be grouped into hard or soft parameter sharing. Hard parameter sharing models share lower layers of the network, while keeping separate output layers for each task (Thrun 1996; Caruana 1997; Nekrasov et al. 2019; Dvornik et al. 2017; Bilen and Vedaldi 2016; Pentina and Lampert 2017; Doersch and Zisserman 2017; Rudd, Günther, and Boult 2016). In contrast, soft parameter sharing provides each task with its own network branch (Long et al. 2017; Misra et al. 2016; Dai, He, and Sun 2016; Tessler et al. 2017; Lu et al. 2017; Zhou et al. 2020), where information sharing is achieved by adding intermediate modules (Long et al. 2017; Misra et al. 2016). Some MTL methods focus on weighing different task-specific loss functions, where manually tuning static loss weights has improved performance (Sermanet et al. 2014; Kokkinos 2017; Eigen and Fergus 2015; Zhang et al. 2018). This is because some tasks may require more attention than others. However, manually tuning loss weights is not scalable. Dynamic weighting schemes have become popular due to their scalability and equal effectiveness (Guo et al. 2018; Li et al. 2017; Kendall, Gal, and Cipolla 2018; Liebel and Körner 2018; Liu, Johns, and Davison 2019; Groenendijk et al. 2021; Chen et al. 2018; Sener and Koltun 2018), and can be combined with other architecture modifications (Liu, Johns, and Davison 2019). We study two weighting methods (Kendall, Gal, and Cipolla 2018; Groenendijk et al. 2021), which we discuss in Sec. 3.2.

Different from MTL, **multi-domain learning (MDL)** seeks to learn the same types of tasks (e.g., classification), but each task is composed of a different domain (Bilen and Vedaldi 2017). MDL models seek to achieve high performance on all tasks. Here, we define each task as a separate domain. During inference, an MDL model is given a data sample with an associated domain label and it performs only the task associated with the indicated domain. Different from MTL, the feature extractor is required to capture different input distributions in the same feature space. Like MTL models, MDL models fall into the hard (Bilen and Vedaldi 2017; Rebuffi, Bilen, and Vedaldi 2017, 2018; Berriel et al. 2019; Mallya, Davis, and Lazebnik 2018; Bulat et al. 2020; Mancini et al. 2018) or soft (Rosenfeld and Tsotsos 2018; Guo et al. 2019; Li and Vasconcelos 2019; Lee, Stokes, and Eaton 2019) parameter sharing categories. However, MDL models must align the feature spaces of multiple domains that contain different lower-level features. Thus, hard sharing is often not performed strictly, but rather by adapting model weights according to domain-specific adaptor modules (Rebuffi, Bilen, and Vedaldi 2017, 2018) or learning domain-specific weight masks (Mallya, Davis, and Lazebnik 2018; Berriel et al. 2019; Mancini et al. 2018). Techniques are developed to also enhance feature sharing across domains (Rebuffi, Bilen, and Vedaldi 2017, 2018; Rosenfeld and Tsotsos 2018). While most MDL studies seek to design models that facilitate "transfer," defined solely by performance improvement, empirical studies demonstrating the contribution of different factors to performance are missing.

MTL and MDL are closely related to continual learning, which requires models to sequentially learn new tasks (Parisi et al. 2019; Mirzadeh et al. 2020; Van de Ven and Tolias 2019), instead of jointly learning them. MTL and MDL are also related to transfer learning (Zhuang et al. 2020), which aims to improve a model's performance on a target task after pretraining on a source task. Knowledge learned during pretraining can benefit the model in learning the target task. Domain adaptation (Wang and Deng 2018) is a subcategory of transfer learning where the input domain changes and the model must maintain performance on different domains.

2.2 Knowledge Transfer

Knowledge transfer is a primary goal in MTL and MDL. An MTL or MDL model exhibits positive transfer when the jointly trained network outperforms a network trained independently on the corresponding task. Based on this phenomenon, Liebel and Körner (2018) introduced auxiliary tasks, which have been shown to be effective in boosting performance on the main task at a low cost of task construction.

Knowledge transfer to the target task is the main objective in transfer learning. There have been studies examining the relationship between MTL performance and task transfer in a transfer learning setting (Zamir et al. 2018; Dwivedi and Roig 2019). However, Standley et al. (2020) found that there was no correlation between MTL performance and transfer learning performance. Knowledge transfer has also been studied in continual learning, where a model's ability to perform better on previous tasks (backward transfer) and unseen tasks (forward transfer) is measured (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018; Díaz-Rodríguez et al. 2018). More generally, Standley et al. (2020) performed an empirical analysis of transfer in MTL by proposing methods to find the best performing task combinations under a specified computation budget. Here, we go beyond simple measures of network performance to study transfer by introducing metrics to capture interference and transfer separately. Further, the MDL setting

allows us to study the role task similarity plays in transfer.

3 Experimental Protocols

3.1 Problem Formulation

We provide a model with a set of classification tasks $\{\mathcal{T}_t\}_{t\leq T}$, along with a set of domains $\{\mathcal{D}_t\}_{t\leq T}$, where $t \leq T$ is a task label. Each task contains a set of N_t samples, $\{(\mathbf{x}_i, y_i, t)\}_{i=1}^{N_t}$, where $\mathbf{x}_i$ is an image, y_i is the corresponding label, and t is the task label. Our models fall into the hard parameter sharing family of MDL models, where all tasks share the same base model parameters $\boldsymbol{\theta}_{share}$, but contain separate output heads $\boldsymbol{\theta}_{1\cdots T}$ for tasks $\mathcal{T}_{1\cdots T}$, respectively. All tasks are trained jointly. We optimize the set of model parameters, $(\boldsymbol{\theta}^*_{share}, \boldsymbol{\theta}^*_{1\cdots T})$, by minimizing the following loss:

$$\mathcal{L}_{total}\left(\boldsymbol{\theta}_{share}, \boldsymbol{\theta}_{1\cdots T}\right) = \sum_{t\leq T} \lambda_t \mathcal{L}_t\left(\boldsymbol{\theta}_{share}, \boldsymbol{\theta}_t\right) \quad , \qquad (1)$$

where $\mathcal{L}_t(\boldsymbol{\theta}_{share}, \boldsymbol{\theta}_t)$ is the loss for task t and λ_t is the weight on $\mathcal{L}_t$ which regulates the importance of each task during training. Note that for each sample, we provide task labels to the models during calculation of the task-specific losses, so $\mathcal{L}_t$ is only calculated based on outputs from $\boldsymbol{\theta}_t$ without any information from other classification heads.

3.2 Multi-Domain Loss Weighting Methods

We investigate the effectiveness of widely-used loss weighting schemes on task losses $\mathcal{L}_{1\cdots T}$, where each $\mathcal{L}_t$ is a standard cross-entropy loss. While these loss weighting schemes have mostly been studied in MTL, we extend them to the MDL setting and describe them below.

Uniform – In Uniform weighting, we use equal task weights, i.e., λ_t is set to 1.

Uncertainty – The uncertainty method (Kendall, Gal, and Cipolla 2018) models multiple classification objectives with softmax likelihoods by introducing task-dependent uncertainty. Intuitively, this metric captures the relative confidence between tasks, is simple to implement, and is a popular baseline in MTL literature. Task weights λ_t are defined as $1/\varepsilon_t^2$ where $\varepsilon_{1\cdots T}$ are learnable noise parameters. Following (Liebel and Körner 2018), we add $\sum_{t\leq T} \log\left(1+\varepsilon_t^2\right)$ to the loss to avoid trivial solutions.

Coefficient of Variations (CoV) – CoV is a recent method that achieves state-of-the-art performance in single-task multi-loss settings (Groenendijk et al. 2021). Loss weights are estimated based on the variance of single-task loss values in relation to their mean. We use the full history of loss statistics for calculations as described in (Groenendijk et al. 2021). Because CoV's loss weights sum to one, we multiply them by T to account for scale differences with other methods.

3.3 Datasets

We perform experiments on three natural image datasets - CIFAR-100 (object/scenery images) (Krizhevsky and Hinton 2009), MiniPlaces (scenery images) (Zhou et al. 2017), and Tiny-ImageNet (object/scenery images) (Le and Yang 2015). Prior studies have shown that transfer is sensitive to the number of samples in each task (Wu, Zhang, and Ré 2019; Hernandez et al. 2021), so we fix all domains to have an equal number of samples and classes to focus on other factors influencing transfer. Specifically, we use all categories from CIFAR-100 and MiniPlaces and randomly sample a fixed set of 100 categories from Tiny-ImageNet. We then sample a fixed set of 500 training images for each MiniPlaces category. We resize images from MiniPlaces and Tiny-ImageNet to 32×32 pixels using bicubic interpolation. We use all test samples pertaining to chosen categories in evaluation. Previous MDL studies (Bilen and Vedaldi 2017; Rebuffi, Bilen, and Vedaldi 2017) perform round-robin batch training, where each batch comes from a single dataset. We instead combine all datasets for training, meaning each batch could contain samples from any dataset.

3.4 Metrics

Performance: In prior MDL studies, transfer is quantified as the raw accuracy gain as compared to a single-task baseline (Zamir et al. 2018; Dwivedi and Roig 2019; Standley et al. 2020). This metric is insufficient since performance gain consists of both *transfer* and *interference*. In this setting, we cannot infer whether performance gain coincides directly with transfer, so we introduce new metrics.

Suppose we have a model, $\mathcal{M}_{\mathcal{T}_t}$, trained on a single domain, $\mathcal{T}_t$. Given a test set $\mathcal{D}_{test}$, we denote the set of samples predicted correctly by $\mathcal{M}_{\mathcal{T}_t}$ as $\mathcal{D}_{test}^{correct} \subseteq \mathcal{D}_{test}$ and the samples predicted incorrectly as $\mathcal{D}_{test}^{incorrect} = \mathcal{D}_{test} \setminus \mathcal{D}_{test}^{correct}$. Suppose that an MDL model makes k and k' correct predictions on $\mathcal{D}_{test}^{correct}$ and $\mathcal{D}_{test}^{incorrect}$, respectively. We define the following metrics to examine its performance on $\mathcal{T}_t$:

$$\textbf{PerfGain} = \frac{k + k' - |\mathcal{D}_{test}^{correct}|}{|\mathcal{D}_{test}|} \times 100\% \quad , \qquad (2)$$

$$\textbf{Interference} = \frac{|\mathcal{D}_{test}^{correct}| - k}{|\mathcal{D}_{test}^{correct}|} \times 100\% \quad , \qquad (3)$$

$$\textbf{Transfer} = \frac{k'}{|\mathcal{D}_{test}^{incorrect}|} \times 100\% \quad . \qquad (4)$$

Performance Gain (PerfGain) is similar to raw performance change metrics used by previous work, but it is measured relative to the single-domain model's performance. Our transfer score, which is the percentage of samples the MDL model answers correctly among those that the single-domain model fails to solve, indicates how much performance improves by jointly training with other domains. Conversely, interference computes the percentage of samples that the MDL model forgets how to solve among those that the single-domain model answers correctly. It measures the amount of performance degradation due to joint training.

Task Similarity: We hypothesize that task similarity in the natural world plays a role in transfer. Since MDL models learn shared feature spaces to simultaneously perform well on many tasks, we calculate task similarity based on feature similarity. That is, given two tasks, $\mathcal{T}_1$ and $\mathcal{T}_2$, we compute the representational similarity between the associated single-domain models $\mathcal{M}_{\mathcal{T}_1}$ and $\mathcal{M}_{\mathcal{T}_2}$. We evaluate the output of the two models on a fixed dataset and obtain two sets of representations, X_1 and X_2. This dataset consists of

50 test samples from each class in each dataset used in our experiments. We define the similarity between $\mathcal{T}_1$ and $\mathcal{T}_2$ at a particular capacity w as $\text{Sim}_w(\mathcal{T}_1, \mathcal{T}_2)$ using the linear Centered Kernel Alignment (CKA) (Kornblith et al. 2019) score between X_1 and X_2. CKA is a popular metric that has demonstrated robustness.

3.5 Network Architectures

We use the ResNet-32 architecture (He et al. 2016) for all experiments. We study the effect of network capacity on knowledge transfer by gradually increasing the percentage of neurons used at each layer (i.e., network width), with the exception of prediction heads. We use four widths in our experiments, with $0.25\times$, $0.5\times$, $1\times$, and $2\times$ of the neurons present at each layer. These models contain 29K, 116K, 463K, and 1848K parameters respectively. In initial studies, ResNet-32 models wider than $2\times$ wide suffered from overfitting, so we do not study them.

Previous work often uses a pretrained model for MDL experiments (Rebuffi, Bilen, and Vedaldi 2017, 2018; Berriel et al. 2019; Mancini et al. 2018). However, this makes it difficult to study transfer in isolation since the model could potentially have informed knowledge about new domains from pretraining. For this reason, we train models from scratch to study transfer. In addition to jointly trained MDL models, we also train an *independent* (single-domain) model separately on each task as a baseline. We provide implementation details and parameter settings for our experiments in Sec. A.

4 Results

The final accuracies for each independent model are in Sec. D. We train one MDL network on each pair of tasks (domain pairings) for all three datasets. We repeat this process for three trials (with different random initializations) and perform analysis on the results. This leads to **36 independent models** and 3 (trials) $\times$ 4 (widths) $\times$ 3 (task groupings) $\times$ 3 (loss weightings) $=$ **108 2-task models**. We compute metrics for each MDL model on each domain. While studying models trained on two domains enables us to analyze the contribution of task similarity more easily, we include the results of **36 3-task models** in Sec. B.1, Sec. B.2, and Sec. E. We organize our results by first outlining high-level questions and claims regarding different factors that could influence interference and transfer and then providing results to address them. Error bars in figures denote standard errors.

4.1 Correlation Between Transfer Learning and MDL Performance

During domain transfer learning, where both the input and label distributions change, a model pretrained on the source domain is optimized towards the target domain alone. The effect of the source domain should remain if we train a model on both domains jointly. Standley et al. (2020) found that the performance of standard transfer learning and MTL are not correlated. Curious to see whether the same conclusion holds in MDL, we perform similar experiments more comprehensively with models with different capacities. We first define *directed* task relationship for domains $B \rightarrow A$ as the change in performance on A as a consequence of either pretraining (in transfer learning) or jointly training (in MDL) on B and *undirected* task relationship between A and B as the average of both directions.

Table 1: Pearson correlation between transfer learning and MDL models. We underline correlations that are not statistically significant at a 95% confidence level.

Capacity	PerfGain	Transfer	Interference
29K	$\underline{0.025}$	0.979	0.842
116K	0.559	0.990	0.982
463K	0.505	0.981	0.982
1848K	0.739	0.987	0.994

Claim *Task relationships in transfer learning positively correlate with that in MDL when characterized by transfer and interference.*

We calculated the directed task relationship on our metrics between all domain pairs at each capacity, under transfer learning and MDL settings and provide undirected results in Sec. C. We then check the correlation between the matching relationships in the two settings and include the results in Table 1. For PerfGain, besides the smallest model, there are significant positive correlations between transfer learning and MDL. This means that the source domain that gives good performance to the target domain in transfer learning is likely to remain beneficial to it if they are trained jointly, especially for large models. This is helpful because transfer learning is less time-consuming than MDL.

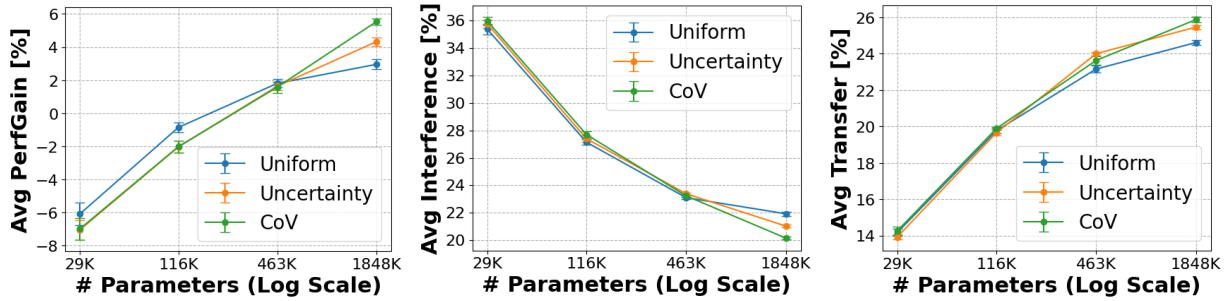
More interestingly, the correlations on transfer or interference are also positive and much stronger, consistent across model sizes. This shows that our sample-wise metrics are more robust and reveal information hidden under the simple accuracy gain. Moreover, such strong correlations allow us to predict the amount of transfer and interference of 2-domain MDL models based on the corresponding transfer learning performance alone.

4.2 What is the role of dynamic loss weighting?

Given the success of loss weighting methods for MTL, we want to examine how well they work in the MDL setting, where the input distribution differs for each task. In this section, we focus on overall performance metrics and discuss task similarity metrics in Sec. 4.5.

Claim *With enough capacity, MDL network performance is improved via loss weightings due to a reduction in interference and increase in transfer across tasks.*

Fig. 2 shows our metric scores for each network capacity using each loss weighting method, averaged across models. From Fig. 2a, we see that dynamic loss weightings only help MDL models for the largest network capacity. This is validated by a paired t-test at each capacity on each pair of the three loss weighting methods. For instance, to perform the test for the Uniform and CoV models, we use PerfGain scores by all Uniform models from all three trials and pair the values with those of the CoV models. For a significance



(a) PerfGain (b) Interference (c) Transfer

Figure 2: (a) PerfGain, (b) Interference, and (c) Transfer scores averaged over the 2-domain networks at each width, plotted as a function of the network's log number of parameters.

level of $\alpha = 0.05$, we found that the loss weighting methods were only statistically significantly different from one another on models with 1848K parameters, where the CoV method outperforms both the Uniform ($p = 0.0165$) and Uncertainty ($p = 0.0456$) methods. The dynamic loss weightings are able to reduce interference (Fig. 2b) and increase transfer (Fig. 2c) at the largest capacity, as verified by a similar paired t-test. For smaller capacities, all three methods have similar interference and transfer scores. We also observe that models using Uniform weighting consistently have a negative PerfGain score on CIFAR-100, while the Uncertainty and CoV methods have positive PerfGain scores when using large network capacities (463K and 1848K parameters). We include scores of individual models in Sec. E.

For remaining sections of the paper, we only show results for models using the CoV loss weighting since it performed the best, unless specified otherwise. Results for the Uniform and Uncertainty weightings follow similar trends and are included in Sec. E.

4.3 Relationship Between Interference, Transfer, and Performance

MDL models seek to learn a general feature space that is aligned with each of the single-domain model feature spaces. By aligning the general feature space with individual feature spaces, we expect to see the following: (1) the general feature space captures individual task characteristics, which reduces interference, and (2) more information from individual spaces can be adapted to solve other tasks, which increases transfer. However, this requires verification because it is possible that a large model uses largely disjoint sets of parameters to learn each task, which would yield little transfer and little interference. We refute this claim by observing linear relationships between transfer and interference (with different slopes in each setting; see Fig. 7). Note that the two scores are calculated on completely disjoint sets of samples.

Claim *Interference and transfer metrics complement each other.*

We plot the performance gain, interference, and transfer metrics as a function of network capacity when evaluated on Tiny-ImageNet in Fig. 3. Consistent across all models, the smallest network exhibits negative PerfGain, but non-zero transfer. Conversely, the largest network has positive PerfGain and non-zero interference. This means that transfer exists when there is performance degradation and interference exists when there is performance improvement. Metrics that only capture overall performance fail to show this.

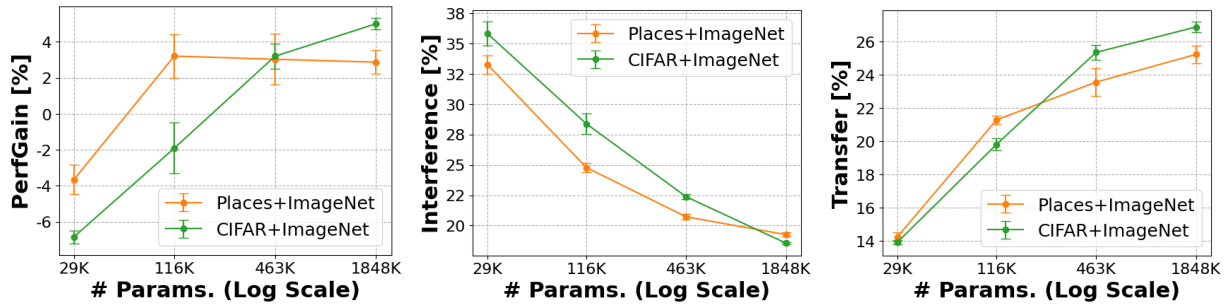
For networks with 29K parameters, joint training with MiniPlaces shows larger PerfGain compared to CIFAR-100. We cannot infer that it comes from task interference rather than transfer, unless looking at the other two metrics. Further, the large standard errors of PerfGain prevent us from making definitive conclusions, while our interference and transfer metrics are calculated in a sample-wise fashion and have smaller errors. These observations demonstrate the importance of disentangling the two metrics from overall performance. Further, the discrepancy between these metrics in Sec. 4.1, Sec. 4.4, and Sec. 4.5 reiterate this claim.

4.4 What is the role of network capacity?

Intuitively, an MDL network with more capacity can allocate more space for storing task-specific information, causing less interference. Conversely, more transfer is expected because larger models can learn how to best share domain-specific information. However, as mentioned in Sec. 4.3, it is possible for a large model to not exhibit transfer or interference at all. Wu, Zhang, and Ré (2019) show that if the capacity of an MTL model is too large then performance might degrade. In MDL, the representations learned from different distributions are expected to be more diverse.

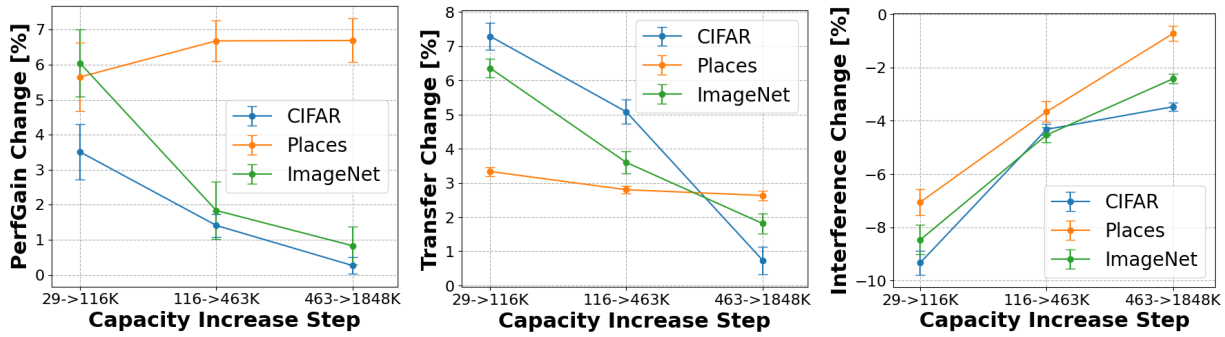
Claim *Greater capacity leads to less interference and more positive transfer.*

Recall that in Fig. 3 we plot our three metrics on Tiny-Imagenet for each pairing. We see that the claim is generally true for all models (Fig. 3, 11, 12, 13). Our findings in the MDL setting align with those of Wu, Zhang, and Ré (2019) in very few cases. For example, models having more than 116K parameters trained on MiniPlaces and Tiny-ImageNet experience a slight decrease in PerfGain on Tiny-ImageNet. However, this task pair still has monotonically decreasing interference and increasing transfer. This can be attributed to the fact that resolving interference plays a more important role in overall performance as capacity increases. Next, we examine transfer on different domains.



(a) PerfGain (b) Interference (c) Transfer

Figure 3: Our evaluation metrics tested on Tiny-ImageNet.



(a) PerfGain (b) Transfer (c) Interference

Figure 4: Average change in (a) PerfGain, (b) Transfer, and (c) Interference for each increment where we increase network capacity. We show the results on each domain, averaged across pairings and loss weighting methods.

Claim *The rate of convergence of transfer on a task is decided by its difficulty.*

Ranked by the single-domain model's accuracy, CIFAR-100 is the easiest task, Tiny-ImageNet is the second easiest, and MiniPlaces is the hardest, consistent across widths. Fig. 4 plots the change in each of our metrics at each step of capacity increase. Scores are averaged across domain pairings and loss weighting methods.

First, the transfer gain is always positive, but decreases monotonically as capacity grows (4b). For smaller widths, single-domain networks fail to utilize the space to capture features that are useful to both training and testing distributions, while seeing data from another domain guides the network to learn more general representations. On the other hand, larger networks have space for more features so such guidance for feature selection is less important. Note that transfer gain is always positive because additional domains still contain additional information. Second, the rate of change/convergence of transfer on the three domains (slope of each curve) positively correlates with their single-domain model's accuracy. We cannot make definitive conclusions, but we hypothesize that easier tasks have a higher convergence rate of transfer due to benefit from general features.

We do not see a clear relationship between the rate of change in PerfGain and task difficulty (Fig. 4a). The rate of change of interference is independent of task difficulty as well (Fig. 4c), and is nearly consistent for each domain. The change is monotonically increasing, indicating that the benefit of capacity on interference gradually decreases.

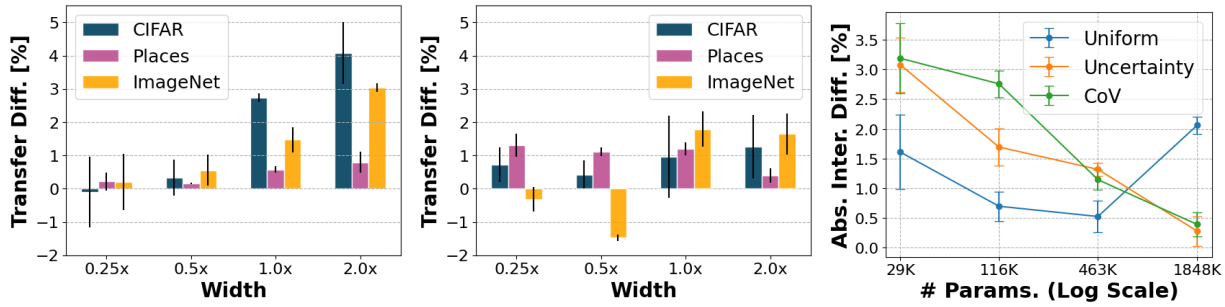
4.5 How does similarity between tasks affect performance?

For each task pair, we compute our similarity metric and see $\text{Sim}_w(\text{CIFAR}, \text{Places}) < \text{Sim}_w(\text{Places}, \text{ImageNet}) < \text{Sim}_w(\text{CIFAR}, \text{ImageNet})$ consistently for all four widths. We now assume this order and show exact scores in Sec. B.3.

In a real-world scenario where we only want to optimize for a particular task while enforcing less strict requirements for performance on the other task, it is important to know whether we should jointly train with another similar or dissimilar task. In a similar task pairing, the representations learned by the single-domain models are more similar, meaning it should be easier to create a general feature space when training the two tasks together. Thus, we would expect that more similar task pairs would benefit the original task more.

Claim *A more similar task pairing yields more transfer, but not necessarily* less interference.

In Fig. 5a and Fig. 5b, we show the amount of transfer gained by using more versus less similar task pairs on each dataset using the Uniform and CoV loss weightings, respectively. A more similar pairing yields more transfer for all the models, violated only by the CoV model's per-



(a) Transfer Diff. (Uniform) (b) Transfer Diff. (CoV) (c) Abs. Interference Diff.

Figure 5: The difference in transfer when using a more similar task pair than a less similar pair using (a) Uniform and (b) CoV weightings. (c) The absolute difference between interference scores given by the two task pairings, averaged across datasets.

formance on Tiny-ImageNet when using small capacities. Consistent across models with at least 1848K parameters, networks trained with more similar pairings have larger PerfGain scores, at most the same amount of interference, and more transfer (see also Fig. 3). Although we find that similarity between tasks strongly affects transfer, similarity also reduces interference and improves PerfGain when we use Uniform weighting, but no relationship emerges when we use dynamic weightings (see Sec. B.3).

Claim *Dynamic loss weightings are more robust to differing amounts of task similarity.*

Fig. 5a shows that, for Uniform weighting, a more similar pair is more beneficial at larger capacities, while Fig. 5b indicates that there is no clear relationship between similarity and capacity when using dynamic loss weighting. We then examine the magnitude of these differences by averaging across their absolute values across the three domains. Fig. 5c examines how the difference between interference changes as capacity increases and shows that networks using loss weighting are less affected by task similarity as size increases.

In summary, dynamic loss weightings yield superior performance at larger capacities (Fig. 2) and, given a domain, reduce the need to determine which other domain to use for joint training. Conversely, if task similarity is known, it is best to choose the most similar domain and use the Uniform weighting model.

5 Discussion and Conclusion

Our empirical results verify that capacity and similarity heavily influence MDL model performance. **We summarize our key takeaways as follows.**

1. Future studies should perform comparisons of MDL models under various model sizes.
2. Transfer and interference are strongly correlated and, along with performance gain, all saturate exponentially with respect to the growth of capacity.
3. When the network capacity is large enough, dynamic loss weighting methods used in MTL are beneficial to MDL models. These benefits include: more performance gain, less interference, more transfer, and more robustness to task similarity level.
4. With enough capacity, choosing more similar domain pairings guarantees more transfer and PerfGain, and at most, the same amount of interference as a less similar pairing, consistent across datasets and loss weighting methods.
5. Our metrics reveal strong correlations between standard transfer learning and MDL models, allowing us to infer MDL performance from transfer learning performance, and vice versa. Correlation results measured using overall accuracy are not consistent across model capacities.

The non-trivial behaviors of MDL models in exhibiting interference and transfer reiterate the importance of our empirical analysis. One future direction our work enables is the study of transfer and interference for additional tasks such as object detection, semantic segmentation, or robotics. While we focused on classification, our metrics are general enough to be applied to other problems. It would be interesting to perform analysis on more challenging datasets such as (Rebuffi, Bilen, and Vedaldi 2017), which involves more dissimilar domains and imbalanced classes. This setting poses further challenges to isolate the impact of each variable on MDL performance. Moreover, we focused on studying transfer in hard parameter sharing MDL models, but future work could examine the role of transfer in soft parameter sharing methods to determine if our conclusions are consistent across MDL techniques. Beyond MDL, it would be interesting to apply our metrics in a continual learning setting (Belouadah, Popescu, and Kanellos 2020; Parisi et al. 2019).

We established experimental protocols and comprehensive metrics for evaluating task interference and knowledge transfer in MDL. We used these protocols to provide the first empirical study of how capacity, task grouping, and dynamic loss weighting contributes to interference and transfer in the MDL regime. Our results indicate that the interplay between these factors is non-trivial and future studies should not assume how one factor affects another. Studying knowledge transfer in the MDL setting will allow researchers to develop more human-like machines with improved performance. These developments could also help researchers generalize transfer to more challenging settings such as open world learning, where an agent must identify new data and then incrementally learn it (Bendale and Boult 2015; Joseph et al. 2021; Xu et al. 2019; Mundt et al. 2020).

Acknowledgements

This work was supported in part by the DARPA/SRI Lifelong Learning Machines program [HR0011-18-C-0051], AFOSR grant [FA9550-18-1-0121], and NSF award #1909696. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements of any sponsor.

References

Belouadah, E.; Popescu, A.; and Kanellos, I. 2020. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*.

Bendale, A.; and Boult, T. 2015. Towards open world recognition. In *CVPR*, 1893–1902.

Berriel, R.; Lathuillere, S.; Nabi, M.; Klein, T.; Oliveira-Santos, T.; Sebe, N.; and Ricci, E. 2019. Budget-aware adapters for multi-domain learning. In *CVPR*, 382–391.

Bilen, H.; and Vedaldi, A. 2016. Integrated perception with recurrent multi-task neural networks. In *NeurIPS*, 235–243.

Bilen, H.; and Vedaldi, A. 2017. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*.

Bulat, A.; Kossaifi, J.; Tzimiropoulos, G.; and Pantic, M. 2020. Incremental multi-domain learning with network latent tensor factorization. In *AAAI*, 10470–10477.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 532–547.

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 794–803. PMLR.

Dai, J.; He, K.; and Sun, J. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 3150–3158.

Díaz-Rodríguez, N.; Lomonaco, V.; Filliat, D.; and Maltoni, D. 2018. Don't forget, there is more than forgetting: new metrics for Continual Learning. In *NeurIPS Workshops*.

Doersch, C.; and Zisserman, A. 2017. Multi-task self-supervised visual learning. In *ICCV*, 2051–2060.

Dvornik, N.; Shmelkov, K.; Mairal, J.; and Schmid, C. 2017. Blitznet: A real-time deep network for scene understanding. In *ICCV*, 4154–4162.

Dwivedi, K.; and Roig, G. 2019. Representation similarity analysis for efficient task taxonomy & transfer learning. In *CVPR*, 12387–12396.

Eigen, D.; and Fergus, R. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2650–2658.

Groenendijk, R.; Karaoglu, S.; Gevers, T.; and Mensink, T. 2021. Multi-loss weighting with coefficient of variations. In *WACV*, 1469–1478.

Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; and Fei-Fei, L. 2018. Dynamic task prioritization for multitask learning. In *ECCV*, 270–287.

Guo, Y.; Li, Y.; Wang, L.; and Rosing, T. 2019. Depthwise convolution is all you need for learning multiple visual domains. In *AAAI*, 8368–8375.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hernandez, D.; Kaplan, J.; Henighan, T.; and McCandlish, S. 2021. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456. PMLR.

Joseph, K.; Khan, S.; Khan, F. S.; and Balasubramanian, V. N. 2021. Towards open world object detection. In *CVPR*, 5830–5840.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 7482–7491.

Kokkinos, I. 2017. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 6129–6138.

Kornblith, S.; Norouzi, M.; Lee, H.; and Hinton, G. 2019. Similarity of neural network representations revisited. In *ICML*, 3519–3529. PMLR.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7: 7.

Lee, S.; Stokes, J.; and Eaton, E. 2019. Learning Shared Knowledge for Deep Lifelong Learning using Deconvolutional Networks. In *IJCAI*, 2837–2844.

Li, C.; Yan, J.; Wei, F.; Dong, W.; Liu, Q.; and Zha, H. 2017. Self-paced multi-task learning. In *AAAI*.

Li, Y.; and Vasconcelos, N. 2019. Efficient multi-domain learning by covariance normalization. In *CVPR*, 5424–5433.

Liebel, L.; and Körner, M. 2018. Auxiliary tasks in multi-task learning. *arXiv preprint arXiv:1805.06334*.

Liu, S.; Johns, E.; and Davison, A. J. 2019. End-to-end multi-task learning with attention. In *CVPR*, 1871–1880.

Long, M.; Cao, Z.; Wang, J.; and Yu, P. S. 2017. Learning multiple tasks with multilinear relationship networks. In *NeurIPS*, 1593–1602.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In *NeurIPS*, 6467–6476.

Lu, Y.; Kumar, A.; Zhai, S.; Cheng, Y.; Javidi, T.; and Feris, R. 2017. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 5334–5343.

Mallya, A.; Davis, D.; and Lazebnik, S. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 67–82.

Mancini, M.; Ricci, E.; Caputo, B.; and Rota Bulo, S. 2018. Adding new tasks to a single network with weight transformations using binary masks. In *ECCV Workshops*.

Mirzadeh, S. I.; Farajtabar, M.; Gorur, D.; Pascanu, R.; and Ghasemzadeh, H. 2020. Linear Mode Connectivity in Multitask and Continual Learning. In *ICLR*.

Misra, D. 2020. Mish: A self regularized non-monotonic neural activation function. In *BMVC*.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *CVPR*, 3994–4003.

Mundt, M.; Hong, Y. W.; Pliushch, I.; and Ramesh, V. 2020. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *arXiv preprint arXiv:2009.01797*.

Nekrasov, V.; Dharmasiri, T.; Spek, A.; Drummond, T.; Shen, C.; and Reid, I. 2019. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. In *ICRA*, 7101–7107.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*.

Pentina, A.; and Lampert, C. H. 2017. Multi-task learning with labeled and unlabeled tasks. In *ICML*, 2807–2816. PMLR.

Qiao, S.; Wang, H.; Liu, C.; Shen, W.; and Yuille, A. 2019. Micro-Batch Training with Batch-Channel Normalization and Weight Standardization. *arXiv preprint arXiv:1903.10520*.

Rebuffi, S.-A.; Bilen, H.; and Vedaldi, A. 2017. Learning multiple visual domains with residual adapters. In *NeurIPS*, 506–516.

Rebuffi, S.-A.; Bilen, H.; and Vedaldi, A. 2018. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, 8119–8127.

Rosenfeld, A.; and Tsotsos, J. K. 2018. Incremental learning through deep adaptation. *IEEE TPAMI*, 42(3): 651–663.

Rudd, E. M.; Günther, M.; and Boult, T. E. 2016. Moon: A mixed objective optimization network for the recognition of facial attributes. In *ECCV*, 19–35. Springer.

Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. In *NeurIPS*, 525–536.

Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; and LeCun, Y. 2014. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *ICML*, 9120–9132.

Tessler, C.; Givony, S.; Zahavy, T.; Mankowitz, D.; and Mannor, S. 2017. A deep hierarchical approach to lifelong learning in minecraft. In *AAAI*.

Thrun, S. 1996. Is learning the n-th thing any easier than learning the first? In *NeurIPS*, 640–646.

Van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. In *NeurIPS Workshops*.

Wang, M.; and Deng, W. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153.

Wu, S.; Zhang, H. R.; and Ré, C. 2019. Understanding and Improving Information Transfer in Multi-Task Learning. In *ICLR*.

Wu, Y.; and He, K. 2018. Group normalization. In *ECCV*, 3–19.

Xu, H.; Liu, B.; Shu, L.; and Yu, P. 2019. Open-world learning and application to product classification. In *WWW*, 3413–3419.

Zamir, A. R.; Sax, A.; Shen, W.; Guibas, L. J.; Malik, J.; and Savarese, S. 2018. Taskonomy: Disentangling task transfer learning. In *CVPR*, 3712–3722.

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018. Context encoding for semantic segmentation. In *CVPR*, 7151–7160.

Zhang, W.; Deng, L.; Zhang, L.; and Wu, D. 2020. Overcoming negative transfer: A survey. *arXiv preprint arXiv:2009.00909*.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6): 1452–1464.

Zhou, L.; Cui, Z.; Xu, C.; Zhang, Z.; Wang, C.; Zhang, T.; and Yang, J. 2020. Pattern-structure diffusion for multi-task learning. In *CVPR*, 4514–4523.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76.

A Implementation Details & Parameter Settings

We replace all BatchNorm (Ioffe and Szegedy 2015) layers in ResNet-32 with GroupNorm (Wu and He 2018). We manually fine-tune the hyper-parameter of GroupNorm (i.e., number of groups) as well as several training hyper-parameters for the highest performance under each setting (i.e., task pairs and widths). We adopt an adaptive grouping strategy similar to (Qiao et al. 2019) such that each GroupNorm layer has $\min\{32, (\text{number of channels})/k\}$ groups. This parameter setting was chosen based on a grid search over $k \in \{2, 4, 8, 16, 32\}$, as well as using a fixed number of groups and searching over $\{1, 4, 8, 16, 32\}$. The adaptive grouping with $k = 2$ consistently yielded the best performance over all capacities and task pairs, so we kept it fixed for all experiments.

We also replace all ReLU activation functions with the Mish activation function (Misra 2020) to prevent gradient vanishing. For all of our experiments, we use a batch size of 128 samples and the stochastic gradient descent (SGD) optimizer with initial learning rate of 0.1, momentum of 0.9, and weight decay of 10^{-4}. For the single-domain networks, we multiply the learning rate by 0.1 at epochs 140 and 210, and train the model for 250 epochs. For the multi-domain experiments, we multiply the learning rate by 0.1 at epochs 150 and 250, and train the model for 300 epochs. The learning rate was chosen based on a grid search over the following values: $\{0.01, 0.05, 0.1, 0.15, 0.2\}$.

B Additional Analysis

B.1 Summary Plots for 3-Domain Models

Similar to Fig. 2, we plot the average PerfGain, Interference, and Transfer curves as a function of network capacity using 3-domain models (models trained jointly on CIFAR-100, MiniPlaces, and Tiny-ImageNet) in Fig. 6. The general trend in the 2-domain results – that PerfGain and Transfer increases, and Interference decreases as capacity grows – holds true for the 3-domain models. However, the 3-domain models' scores saturate slower because training on more domains requires more network capacity to perform well. Unlike the 2-domain PerfGain curve, dynamic loss weighting does not seem to help performance of 3-domain models with 1848K parameters in general.

B.2 Relationship Between Transfer and Interference

In Fig. 7 (2-domain) and Fig. 8 (3-domain), we plot each network's transfer as a function of interference using each of the three loss weighting methods. We observe that interference and transfer exhibit a linear relationship, which is consistent across methods. As mentioned in Sec. 4.3, the correlation strength (slope of the best-fit line) is independent of capacity or domain. Moreover, the correlation strength tells us whether a method focuses more on transfer (more negative slope) or interference (less negative slope). For instance, the best-fit line for CoV has a less negative slope, indicating that it focuses more on transfer.

B.3 Similarity Analysis

Task Pairing Similarity Scores In Sec. 4.5, we use task similarity scores to study interference and transfer. In Table 2, we show the exact CKA similarity scores between each pair of tasks. We observe that the order is consistent across network widths with:

$$\begin{aligned} &\text{Sim}_w(\text{CIFAR}, \text{Places}) < \\ &\text{Sim}_w(\text{Places}, \text{ImageNet}) < \\ &\text{Sim}_w(\text{CIFAR}, \text{ImageNet}) \ . \end{aligned} \tag{5}$$

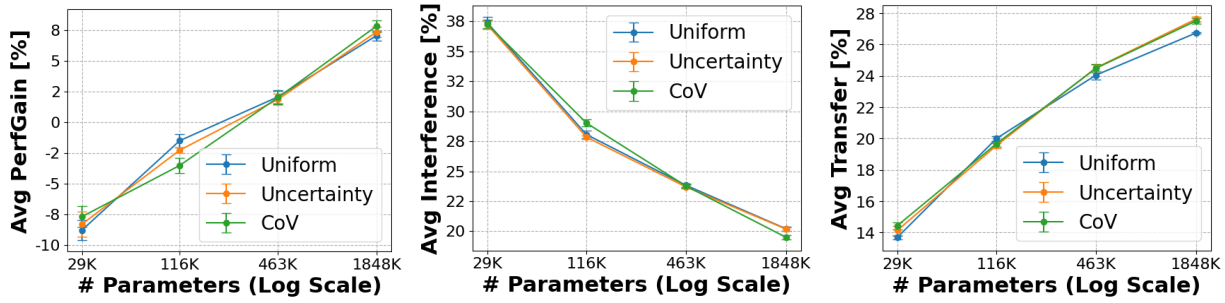
One question that arises is: could the MDL models perform well on each task by only learning task-specific information in the separate classification heads instead of learning generalizable features? Since we have more than one classification head, the shared feature representations would need to be linearly separable in all domains to perform well. This would make it difficult for the model to learn domain-specific features only in the task heads. Further, we verify that the representations learned by the MDL models do not diverge much from those learned by the single domain experts as reflected by CKA scores that were consistently greater than 0.6, which is much higher than the CKA between two single domain experts shown in Table 2.

B.4 Relationship Between Similarity, Capacity, and Loss Weighting

Transfer Analysis In Fig. 9 (middle row), we plot the transfer gain when using a more similar task pairing for 2-domain models. As discussed in the main text, using a more similar task pair with the Uniform method always yields more transfer. For both the Uncertainty and CoV weightings, the conclusion is true except for Tiny-ImageNet at width $0.25\times$ and $0.5\times$, and for CIFAR-100 at width $0.5\times$. These special cases indicate that networks using loss weightings sometimes fail to utilize the similarity between tasks to facilitate transfer, which is likely due to a lack of capacity.

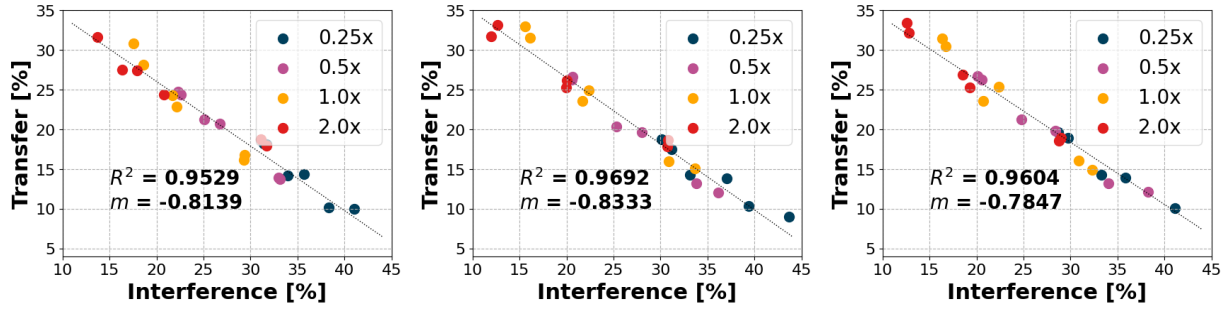
We then examine the magnitude (absolute value) of this difference, averaged across datasets, as shown in Fig. 10b. Using the Uniform weighting, the difference increases as width increases, indicating that it is important to choose the more similar pair when using large capacities to obtain the most transfer. Conversely, using the dynamic loss weighting methods, the absolute difference in transfer between task pairs remains the same regardless of width.

Interference Analysis In Fig. 9 (last row), we plot the interference change when using a more similar task pair for 2-domain models. Interestingly, unlike transfer, we do not see a clear trend when considering if the value of the interference difference is positive or negative. For the Uniform method, the difference is generally negative, indicating that using a more similar task pair generally yields less interference. We do not see a clear relationship between similarity and interference for dynamic loss weightings, where MiniPlaces benefits most from a more similar task pair, while Tiny-ImageNet benefits more from a less similar one. Therefore, jointly training a more similar domain does not necessarily guarantee that there will be less interference than training using a less similar domain. However, we next discuss that similarity does not



(a) PerfGain (b) Interference (c) Transfer

Figure 6: (a) PerfGain, (b) Interference, and (c) Transfer scores averaged over the 3-domain networks at each width, plotted as a function of the network's log number of parameters.



(a) Uniform (b) Uncertainty (c) CoV

Figure 7: Scatter plot of transfer versus interference using (a) Uniform, (b) Uncertainty, and (c) CoV loss weighting with 2-domain models. Each dot corresponds to a network's performance on some domain at some width. The R^2 score is calculated based on the best-fit line, with a slope of m.

have to be considered when using loss weighting methods with large capacity networks.

Using the Uniform loss weighting, there is no clear relationship between the magnitude of the interference differences between task pairs and network capacity (Fig. 10c). Using dynamic weightings, the magnitude of the interference differences decreases as capacity grows, as shown in the main text (Fig. 5c) as well as in Fig. 10c. Therefore, although we find no correlation between similarity and interference for dynamic weightings, we show that the differences are negligible when we have a large network.

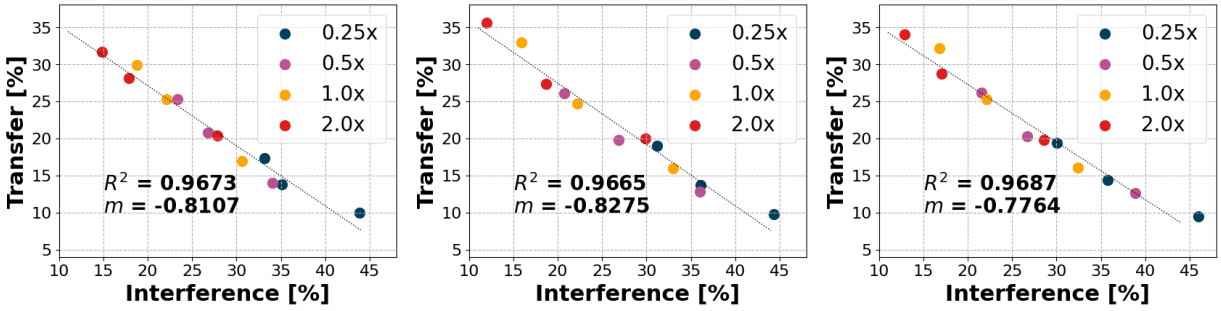
PerfGain Analysis Performance improvement is a result of the interplay between transfer and interference, and we plot the PerfGain change when using a more similar task pair for 2-domain models (first row of Fig. 9). For the Uniform model, the behavior of PerfGain is consistent with transfer, while for the dynamic weighting models, its behavior is consistent with the negation of interference.

In terms of the magnitude of PerfGain differences, dynamic loss weighting methods are less affected by the difference in task similarity as capacity grows (Fig. 10a). On the other hand, the benefit of a more similar task pair grows as capacity grows if we use the Uniform method.

Similar to our discussion in Sec. 4.5, the key takeaway of Fig. 9 and Fig. 10 is as follows: dynamic loss weightings yield superior performance at larger capacities (Fig. 2) and, given a domain, reduce the need to determine which other domain to use for joint training. On the other hand, if the order of similarity is known, it is best to choose the most similar domain and use the Uniform weighting model.

C Transfer Learning and MDL Performance Correlation

We include the full table of directed (Table 4) and undirected (Table 5) domain relationship correlations between Transfer Learning and MDL, including the p-values for statistical significance testing. As discussed in Sec. 4.1, the directed correlation between Transfer Learning and MDL is positive except for PerfGain using the network with smallest capacity. We also see much stronger and statistically significant correlations between the Transfer Learning and MDL relationships using our transfer and interference metrics. Conversely, we do not see correlation between the Transfer Learning and MDL relationship using our PerfGain scores. However, the all correlations using our transfer and interference metrics are statistically significant at a 95% confidence interval, with the exception of our interference score for the network with the smallest capacity. The PerfGain correlations are consistent with the findings of Standley et al. (2020), in which experiments are performed on Transfer Learning and MTL with



(a) Uniform (b) Uncertainty (c) CoV

Figure 8: Scatter plot of transfer versus interference using (a) Uniform, (b) Uncertainty, and (c) CoV loss weighting with 3-domain models. Each dot corresponds to a network's performance on some domain at some width. The R^2 score is calculated based on the best-fit line, with a slope of m.

Table 2: CKA scores of individual models trained on different task pairings at each width of ResNet-32. Scores are listed from the most dissimilar pairings (top) to the most similar pairings (bottom). The mean and standard deviation of CKA scores are reported based on 3 trials with different random initializations.

Task Pairing	$\mathbf{0.25\times}$	$\mathbf{0.5\times}$	$\mathbf{1\times}$	$\mathbf{2\times}$
CIFAR & Places	$0.38(\pm 0.035)$	$0.37(\pm 0.147)$	$0.42(\pm 0.010)$	$0.38(\pm 0.020)$
Places & ImageNet	$0.40(\pm 0.027)$	$0.41(\pm 0.009)$	$0.45(\pm 0.010)$	$0.40(\pm 0.023)$
CIFAR & ImageNet	$0.48(\pm 0.007)$	$0.50(\pm 0.006)$	$0.52(\pm 0.008)$	$0.43(\pm 0.002)$

only a single network (i.e., not varying capacity). As shown in Table 4 and Table 5, correlation results vary across capacity, and the results of experiments that were conducted using only a single capacity may not hold for different network capacities or architectures. Our transfer and interference metrics are more robust for revealing the underlying correlations between Transfer Learning and MDL.

D Single-Domain Model Performance

In Table 3, we show the final accuracy of each single-domain model trained and evaluated on each domain. As mentioned in Sec. 4.4, we define the difficulty of tasks based on their respective single-domain model performance. That is, CIFAR-100 is the easiest and Tiny-ImageNet is the hardest, which is consistent across widths. One observation is that performance gains gradually decrease when adding more network capacity on all three datasets.

E Multi-Domain Model Performances

We compute our evaluation metrics with networks using Uniform, Uncertainty, and CoV loss weighting, as shown in Fig. 11, Fig. 12, and Fig. 13, respectively. We first discuss the results of 2-domain models.

E.1 2-Domain Model Performance

Recall from Sec. 4, there are some general trends that are consistent across domains and methods. First, for small capacities, all networks exhibit negative PerfGain scores, while having positive transfer; at the largest capacity, only the Uniform methods show negative PerfGain on CIFAR-100 and all other networks show positive PerfGain, while having positive interference. Second, at the largest capacity (1848K parameters), all models using a more similar task pairing show larger PerfGain, at most the same amount of interference, and more transfer.

The two dynamic loss weighting methods show very similar behaviors. As mentioned in the main text (Fig. 5c), the interference difference between task pairs decreases as capacity grows when using dynamic loss weighting.

Next, we list cases which indicate that a single overall performance gain metric is not enough to capture both transfer and interference. **(1)** As discussed in Sec. 4.3, using the $0.25\times$ wide CoV networks, we cannot directly infer that a large PerfGain score on Tiny-ImageNet from jointly training with MiniPlaces is attributed to the task pair's ability to reduce interference, rather than increasing transfer (see right column of Fig. 13). **(2)** Similarly, using the $0.25\times$ wide Uniform networks, we cannot directly infer that a large PerfGain on MiniPlaces from jointly training with Tiny-ImageNet is attributed to the task pair's ability to reduce interference, rather than increasing transfer (see middle column of Fig. 11). **(3)** Using the Uniform network jointly trained with MiniPlaces (see left column of Fig. 11), we notice that the PerfGain score on CIFAR-100 decreases at the largest capacity (1848K parameters). Separating the interference and transfer metrics enables us to discover that the network fails to promote more transfer, but still manages to reduce more interference compared to the smaller network at 463K parameters. There are more similar examples in Fig. 11, Fig. 12, and Fig. 13. Our metrics present us with a more comprehensive view of an

Table 3: Performance of independent models at each width of ResNet-32. Mean and standard deviation of accuracy scores are shown based on 3 trials with different random initializations.

Dataset	$\mathbf{0.25}\times$	$\mathbf{0.5}\times$	$\mathbf{1}\times$	$\mathbf{2}\times$
CIFAR-100	$42.40(\pm 0.76)$	$57.12(\pm 0.22)$	$65.27(\pm 0.23)$	$70.48(\pm 0.12)$
MiniPlaces	$23.24(\pm 0.58)$	$29.84(\pm 0.20)$	$32.04(\pm 0.34)$	$32.76(\pm 0.34)$
Tiny-ImageNet	$32.46(\pm 0.63)$	$43.29(\pm 0.74)$	$49.83(\pm 0.76)$	$53.28(\pm 0.31)$

Table 4: Directed domain relationship correlations between Transfer Learning and Multi-Domain Learning models. We report the Pearson's correlation coefficient and the p-value for statistical significance for each experiment.

Capacity	**PerfGain**		**Transfer**		**Interference**	
(# param.)	Pearson's r	p	Pearson's r	p	Pearson's r	p
29K	0.025	0.311	0.979	1×10^{-12}	0.842	1×10^{-5}
116K	0.559	0.016	0.990	5×10^{-15}	0.982	5×10^{-13}
463K	0.505	0.033	0.981	6×10^{-13}	0.982	4×10^{-13}
1848K	0.739	0.001	0.987	4×10^{-14}	0.994	1×10^{-16}

MDL model's performance, as compared to only evaluating overall performance.

E.2 3-Domain Model Performance

Our findings in the 2-domain setting still hold for the 3-domain models. However, 3-domain models exhibit more benefit from the largest capacity because of the need to learn representations general to all three domains. This is reflected by the fact that, as capacity grows, the 3-domain models' PerfGain increases, interference decreases, and transfer increases monotonically in all cases in Fig. 11, Fig. 12, and Fig. 13. Note that at the largest capacity, the 3-domain model always has better performance than the 2-domain model, except on CIFAR-100 using the Uniform weighting. However, we cannot make conclusions on the benefit of more domains because the 3-domain models are trained on more samples.

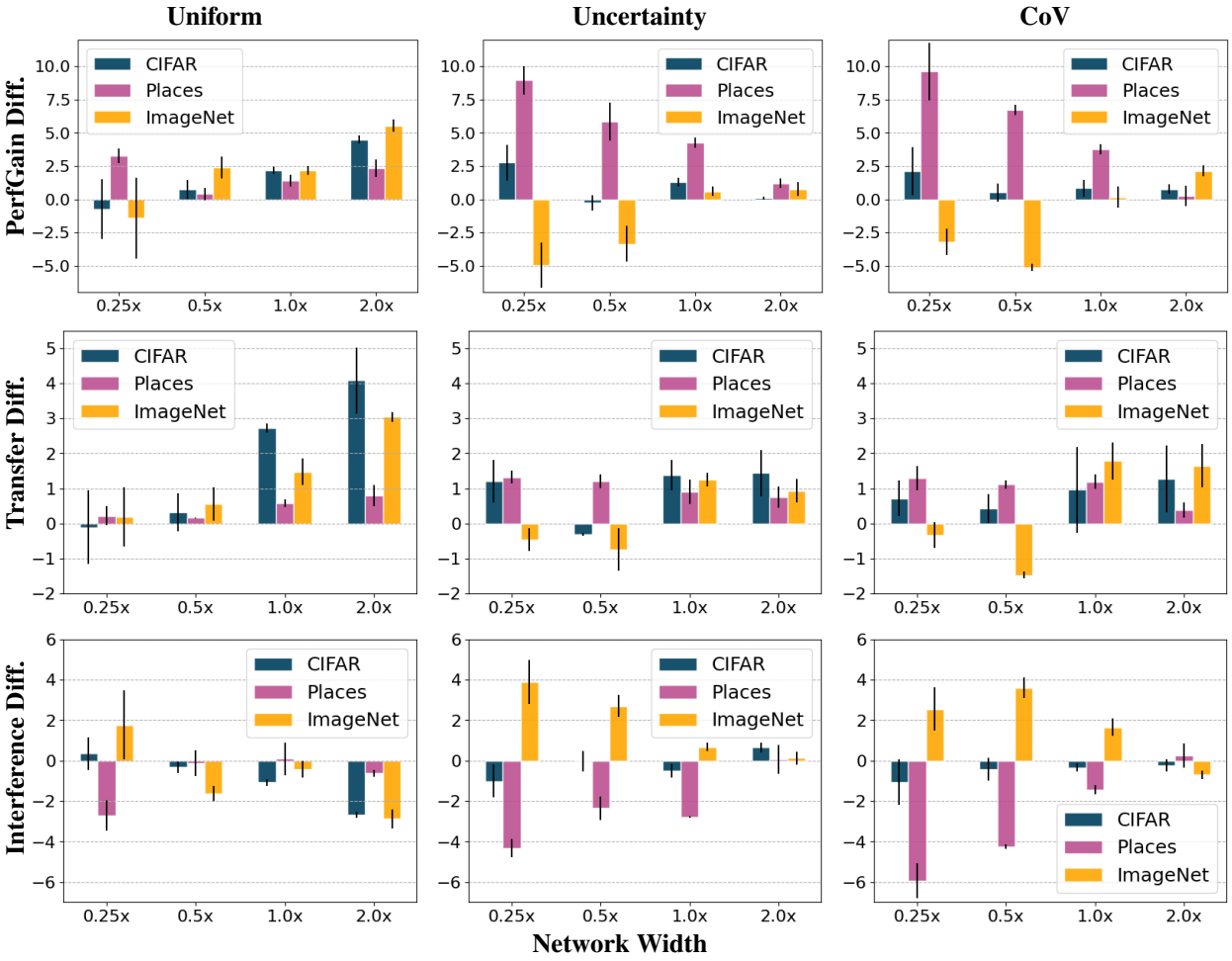
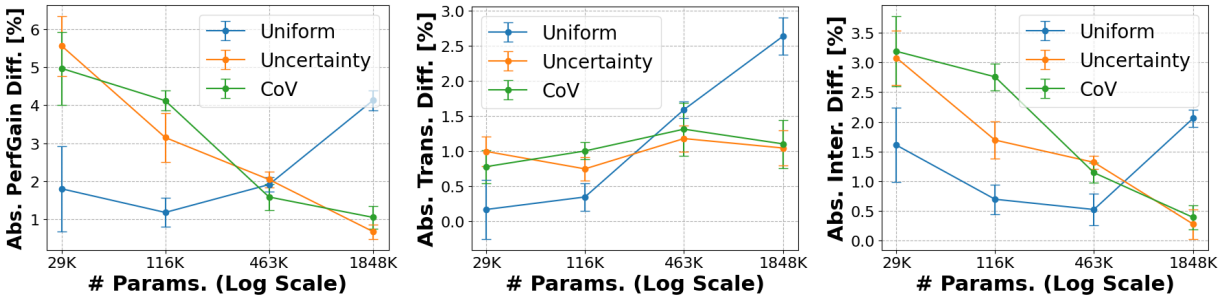


Figure 9: The difference in each metric when using a more similar task pair than a less similar pair. The x-axis is the network width with respect to the original ResNet-32.



(a) PerfGain (b) Transfer (c) Interference

Figure 10: The absolute difference between (a) PerfGain, (b) Transfer, and (c) Interference given by two task pairs averaged across datasets, plotted against the log number of parameters.

Table 5: Undirected domain relationship correlations between Transfer Learning and Multi-Domain Learning models. We report the Pearson's correlation coefficient and the p-value for statistical significance for each experiment.

Capacity	**PerfGain**		**Transfer**		**Interference**	
(# param.)	Pearson's r	p	Pearson's r	p	Pearson's r	p
29K	0.118	0.762	0.942	1×10^{-4}	0.573	0.107
116K	0.651	0.057	0.986	1×10^{-6}	0.922	4×10^{-4}
463K	0.142	0.716	0.962	3×10^{-5}	0.958	4×10^{-5}
1848K	0.222	0.565	0.986	1×10^{-6}	0.987	8×10^{-7}

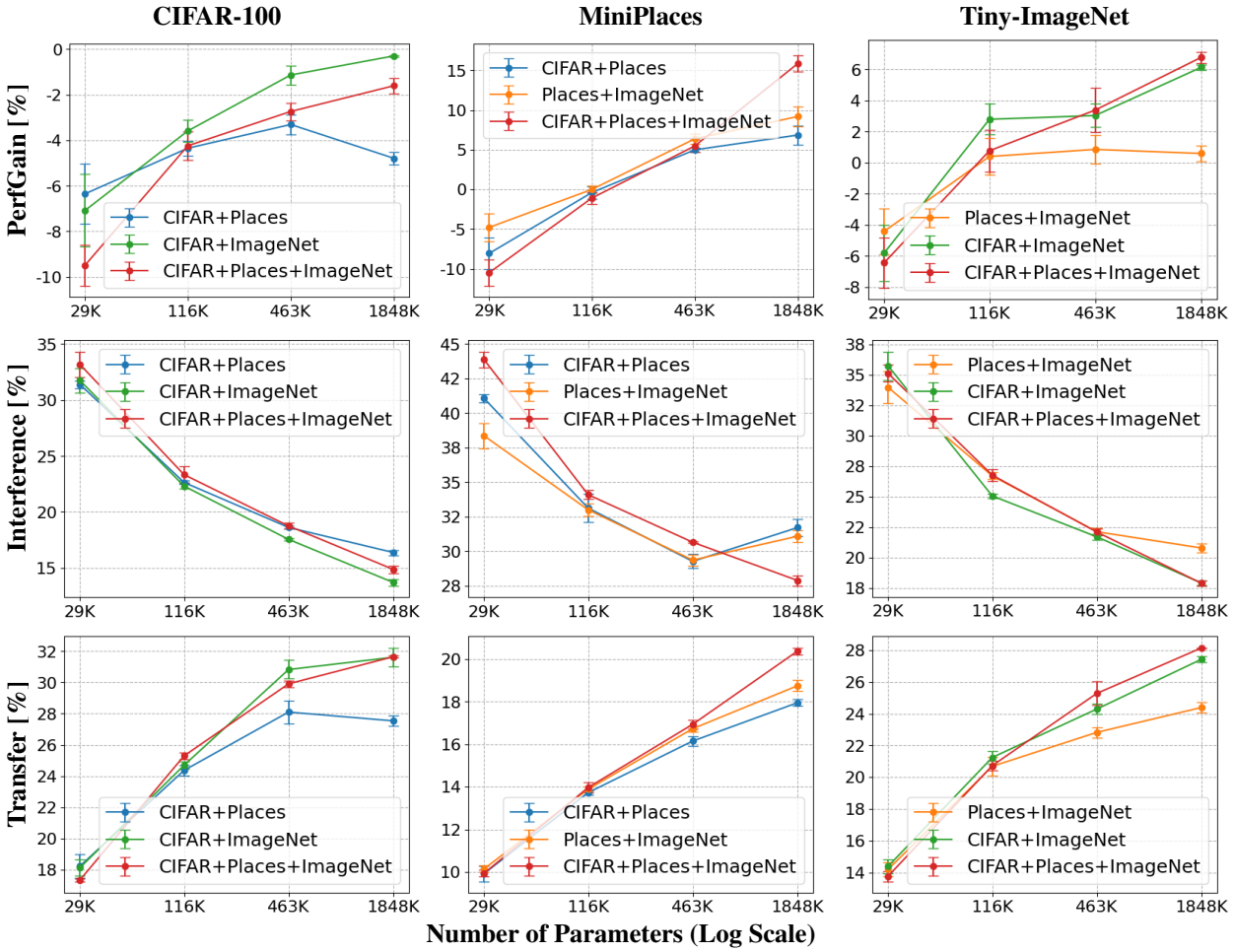


Figure 11: Our evaluation metrics tested with networks using Uniform loss weighting.

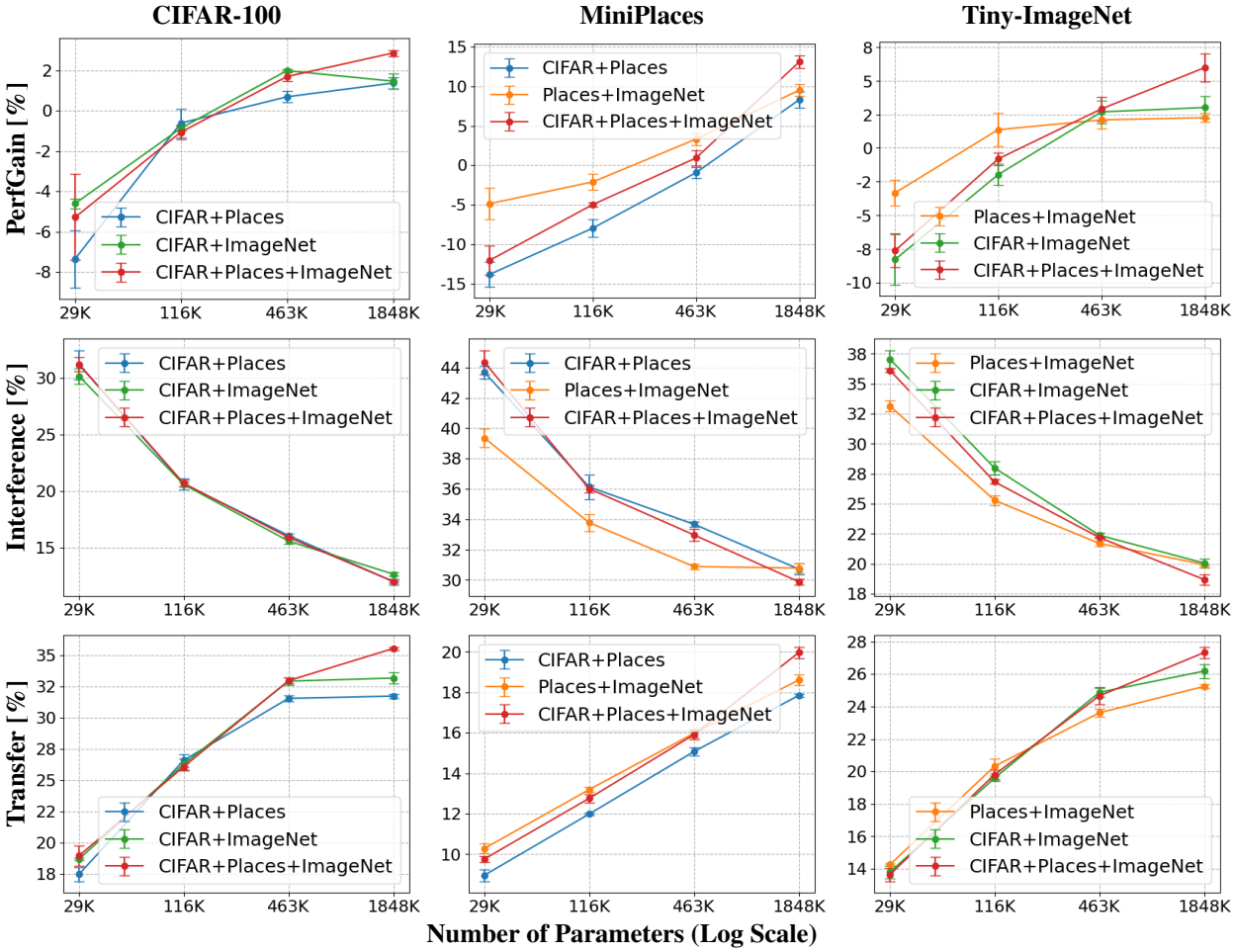


Figure 12: Our evaluation metrics tested with networks using Uncertainty loss weighting.

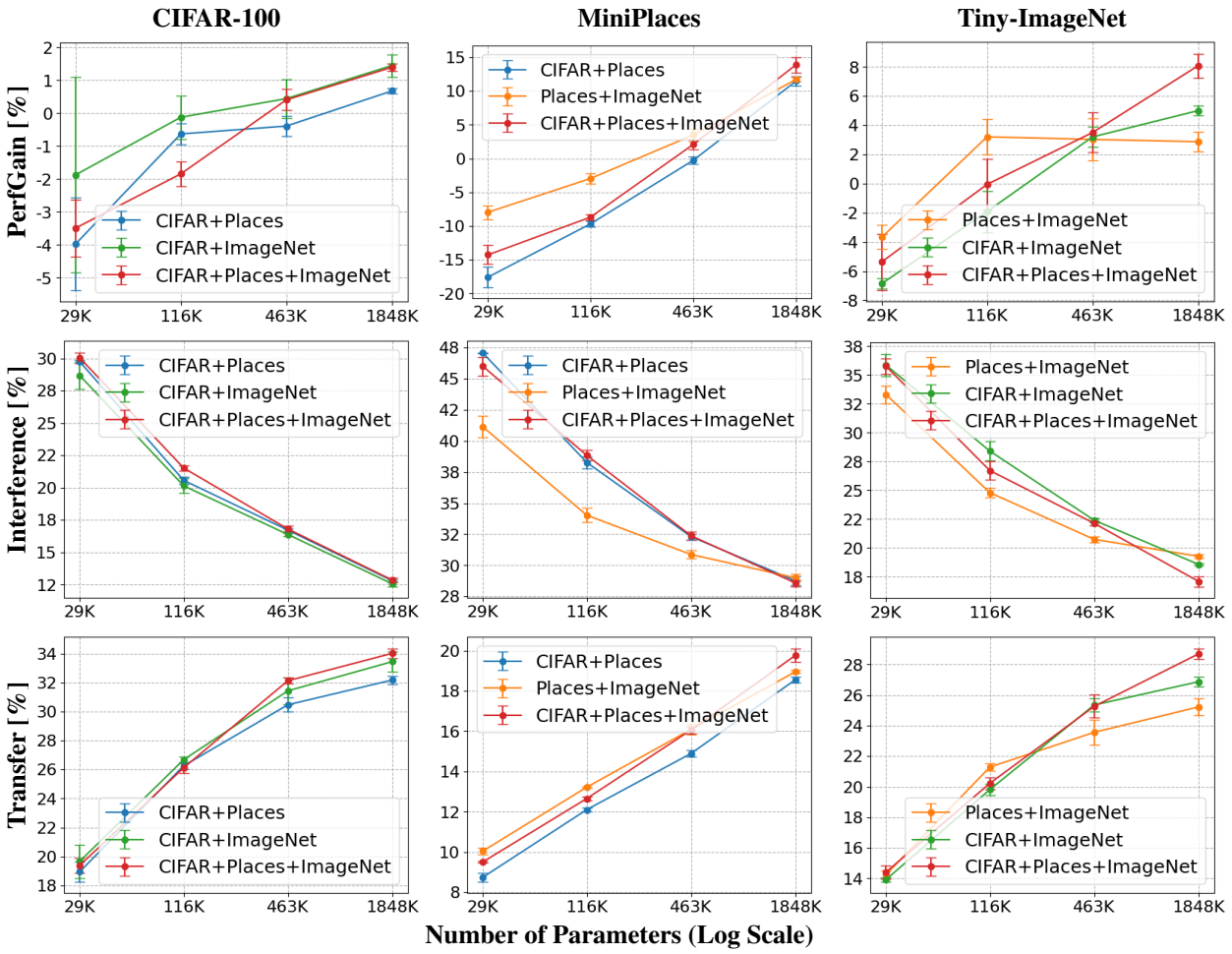


Figure 13: Our evaluation metrics tested with networks using CoV loss weighting. The last column is shown in the main text (Fig. 3).