Data Attribution for Multitask Learning

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Data attribution quantifies the influence of individual training data points on machine learning models, aiding in their interpretation and improvement. While prior work has primarily focused on single-task learning (STL), this work extends data attribution to multitask learning (MTL). Data attribution in MTL presents new opportunities for interpreting and improving MTL models while also introducing unique technical challenges. On the opportunity side, data attribution in MTL offers a natural way to efficiently measure task relatedness, a key factor that impacts the effectiveness of MTL. However, the shared and task-specific parameters in MTL models present challenges that require specialized data attribution methods. In this paper, we propose the *MultiTask Influence Function* (**MTIF**), a novel data attribution method tailored for MTL. MTIF leverages the structure of MTL models to efficiently estimate the impact of removing data points or excluding tasks on the predictions of specific target tasks, providing both data-level and task-level influence analysis. Extensive experiments on both linear and neural network models show that MTIF effectively approximates leave-one-out and leave-one-task-out effects. Moreover, MTIF facilitates fine-grained data selection, consistently improving model performance in MTL, and provides interpretable insights into task relatedness. Our work establishes a novel connection between data attribution and MTL, offering an efficient and scalable solution for measuring task relatedness and enhancing MTL models.

1 Introduction

Data attribution aims to quantify the influence of individual training data points on machine learning models and has been widely used to interpret and improve these models (Koh & Liang, 2017; Hammoudeh & Lowd, 2024). However, most existing literature on data attribution focuses on single-task learning (STL) settings. In contrast, this work explores data attribution in the context of multitask learning (MTL), where multiple related tasks are trained simultaneously to enhance overall performance (Caruana, 1997). Data attribution in MTL presents new opportunities for interpreting and improving MTL, while also introducing distinct technical challenges in comparison to data attribution in STL.

MTL has demonstrated success across a wide range of domains, including computer vision (Zamir et al., 2018), natural language processing (Hashimoto et al., 2017), speech processing (Huang et al., 2015), and recommender systems (Ma et al., 2018). In practice, however, MTL does not always help with the overall performance—training unrelated tasks together often harms the learning performance, a phenomenon known as negative transfer (Standley et al., 2020; Wang et al., 2020; Parisotto et al., 2016; Rusu et al., 2016). As a result, understanding and quantifying task relatedness has become a key focus in MTL research (Ma et al., 2018; Standley et al., 2020; Fifty et al., 2021). Despite this, there is still no consensus on a universally effective and efficient method for measuring task relatedness. In practical applications, practitioners often rely on trial and error—repeatedly

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

training models with different task combinations—as a gold standard to assess task relatedness, a process that is computationally expensive.

Generalizing data attribution methods to MTL offers a promising, efficient, and interpretable way to measure task relatedness in MTL. Many data attribution methods are designed to efficiently approximate the change of model performance when retraining the model with certain data points excluded from the training dataset (Koh & Liang, 2017; Park et al., 2023). Extending these methods to MTL naturally leads to an efficient approximation of the aforementioned trial-and-error process for determining task relatedness. Moreover, data attribution methods allow for fine-grained analysis at the individual data instance level, revealing how data points from one task impact performance on another task. This data-level influence analysis offers more interpretable insights into task relatedness by moving beyond a single metric, providing concrete evidence of how tasks are related through specific data points and their cross-task effects. Please see Appendix A for more related work for data attribution and task relatedness in MTL.

However, MTL introduces unique challenges that require tailored data attribution methods. MTL models typically consist of both shared parameters across all tasks and task-specific parameters for each individual task. When making predictions for a specific task, only a submodel with a subset of the parameters is utilized. As the number of tasks increases, this brings several computational challenges for data attribution. Firstly, since each task corresponds to a separate attribution target, retraining-based data attribution methods (Ghorbani & Zou, 2019; Jia et al., 2019) become prohibitively expensive. Therefore, in this paper, we focus on influence function (IF)-based data attribution methods that do not require repeated retraining. Additionally, tasks in MTL may employ different loss functions, and the number of parameters scales with the number of tasks, further complicating the application of existing IF-based data attribution methods designed for single-task learning (STL). These factors present significant technical and computational challenges when adapting such methods to the MTL setting.

In this paper, we propose the *MultiTask Influence Function* (**MTIF**) to address these challenges. Similar to the IF-based data attribution methods for STL (Koh & Liang, 2017), MTIF leverages a first-order approximation to efficiently estimate the impact of removing a data point from one task on the prediction for another task, without the need for model retraining. Specifically designed for MTL, MTIF derives the influence of data points on the shared and task-specific parameters separately, and exploits the unique structure of MTL models to enhance computational efficiency. MTIF enables the efficient estimation of both data-level and task-level influence, providing a scalable and interpretable solution for data attribution in MTL settings.

We conduct extensive experiments on both linear and neural network models to evaluate the effectiveness of the proposed MTIF. On linear models, the data-level influence scores predicted by MTIF shows a near perfect correlation with the actual change of model outputs obtained by brute-force leave-one-out retraining; the task-level influence estimated by MTIF also strongly correlates with the leave-one-task-out retraining, with an average Pearson correlation around 0.7. On neural network models, the task-level influence estimated by MTIF also shows significant correlation with leave-one-task-out retraining, with Pearson correlation ranging from 0.1 to 0.4. Moreover, the data-level influence estimated by MTIF enables fine-grained data selection for MTL, which demonstrates consistent performance improvements over baselines. Finally, we provide case studies of the most negative data points from one task to another task, providing interpretations about negative transfer.

2 Influence Function for Multitask Data Attribution

In this section, we generalize influence function in STL (see Appendix B) to MTL settings.

2.1 Problem Setup for Multitask Data Attribution

Multitask Learning. MTL aims to solve multiple tasks simultaneously. In many real-world scenarios, tasks are often related and share common underlying structures. MTL leverages shared structures by jointly training tasks to enhance generalization and improve prediction accuracy, especially when tasks are related or when data for individual tasks is limited. A common approach in MTL to facilitate information sharing across tasks is through either soft or hard parameter sharing (Ruder, 2017). In soft parameter sharing, regularization is applied to the task-specific parameters to encourage them to be similar across tasks (Xue et al., 2007; Duong et al., 2015). In contrast, hard parameter sharing learns a common feature representation through shared parameters, while task-specific parameters are used to make predictions tailored to each task (Caruana, 1997). Recently, Duan & Wang (2023) proposed an augmented optimization framework for MTL that accommodates both hard parameter sharing and various types of soft parameter sharing.

We consider a general multitask learning objective that incorporates these common parameter-sharing schemes. Specifically, consider K tasks and for each task $k = 1, \dots, K$, we observe n_k independent samples, denoted by $\{z_{ki}\}_{i=1}^{n_k}$. Let $\ell_k(\cdot;\cdot)$ be the loss function for task k. The MTL objective is given by

$$\mathcal{L}(\boldsymbol{w}) = \sum_{k=1}^{K} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \ell_k(\theta_k, \gamma; z_{ki}) + \Omega_k(\theta_k, \gamma) \right], \tag{1}$$

where $\boldsymbol{\theta} = \{\theta_k \in \mathbb{R}^{d_k}\}_{k=1}^{K}$ are task-specific parameters, $\gamma \in \mathbb{R}^p$ are shared parameters, $\boldsymbol{w} = \{\boldsymbol{\theta}, \gamma\}$ denotes all parameters, and $\Omega_k(\theta_k, \gamma)$ represents the task-level regularization. The parameters are estimated by minimizing (1), i.e., $\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w})$.

Below, we present two special cases of supervised learning within this general framework: one illustrating soft parameter sharing and the other demonstrating hard parameter sharing. Let $z_{ki} = (x_{ki}, y_{ki})$ for $1 \le k \le K$ and $1 \le i \le n_k$, where x_{ki} represents the features and y_{ki} represents the outcomes for the i-th data point in task k.

We provide two concrete examples of MTL models, Multitask Linear Regression with Ridge Penalty (Example 1) and Shared-Bottom Neural Network (Example 2), in Appendix C.

Multitask Data Attribution. In this work, we aim to estimate the contribution of a data point (or a task) to the learning performance on a specific target task $k \in \{1, \dots, K\}$. The performance of any model with parameters (θ_k, γ) on task k can be measured by the average loss over a validation dataset D_k^{v}, i.e, $V_k(\theta_k, \gamma; D_k^{\mathrm{v}}) = \sum_{z \in D_k^{\mathrm{v}}} \ell_k(\theta_k, \gamma; z)/|D_k^{\mathrm{v}}|$. Then the *data-level influence* of the i-th data point from task l on the target task k can be quantified by the following LOO effect:

$$\Delta_k^{li} := V_k(\hat{\theta}_k, \hat{\gamma}; D_k^{\mathrm{v}}) - V_k(\hat{\theta}_k^{(-li)}, \hat{\gamma}^{(-li)}; D_k^{\mathrm{v}}), \tag{2}$$

where $\hat{\theta}_k$ and $\hat{\gamma}$ are from the minimizer of (1) with the full training data, while $\hat{\theta}_k^{(-li)}$ and $\hat{\gamma}^{(-li)}$ are obtained by excluding the data point z_{li} from task l. This data-level attribution metric allows for a fine-grained understanding of the impact each data point from one task has on another task.

Similarly, the *task-level influence* of task l on the target task k is quantified by the leave-one-task-out (LOTO) effect:

$$\Delta_k^{l} := V_k(\hat{\theta}_k, \hat{\gamma}; D_k^{\mathrm{v}}) - V_k(\hat{\theta}_k^{(-l)}, \hat{\gamma}^{(-l)}; D_k^{\mathrm{v}}), \tag{3}$$

where $\hat{\theta}_k^{(-l)}$ and $\hat{\gamma}^{(-l)}$ are obtained by excluding all the data points from task l. The LOTO effect provides a natural and interpretable measure of task relatedness.

2.2 The Proposed Method: Multitask Influence Function

The computational burden of evaluating LOO and LOTO effects becomes even more pronounced in MTL setting compared to STL setting, particularly when the number of tasks is large. To address this challenge, we extend the IF-based approximation to LOO and LOTO effects in MTL. This approach builds on the similar idea of using infinitesimal perturbations on the weights of data points to approximate the removal of individual data points. Specifically, we consider the following data-level $\boldsymbol{\sigma}$-weighted version of the general objective function in (1):

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma}) = \sum_{k=1}^{K} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \sigma_{ki} \ell_{ki}(\theta_k, \gamma) + \Omega_k(\theta_k, \gamma) \right], \tag{4}$$

where $\ell_{ki}(\cdot)$ is shorthand for $\ell_k(\cdot; z_{ki})$. For each weight vector $\boldsymbol{\sigma}$, we solve $\boldsymbol{w}(\boldsymbol{\sigma}) = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$. We propose to use the partial derivative with respect to σ_{li}, i.e.,

$$\frac{\partial V_k(\hat{\theta}_k(\boldsymbol{\sigma}), \hat{\gamma}(\boldsymbol{\sigma}); D_k^{\mathrm{v}})}{\partial \sigma_{li}} \bigg|_{\boldsymbol{\sigma}=\mathbf{1}} = \nabla_\theta V_k(\hat{\theta}_k, \hat{\gamma}; D_k^{\mathrm{v}}) \cdot \frac{\partial \hat{\theta}_k(\boldsymbol{\sigma})}{\partial \sigma_{li}} \Big|_{\boldsymbol{\sigma}=\mathbf{1}} + \nabla_\gamma V_k(\hat{\theta}_k, \hat{\gamma}; D_k^{\mathrm{v}}) \cdot \frac{\partial \hat{\gamma}(\boldsymbol{\sigma})}{\partial \sigma_{li}} \Big|_{\boldsymbol{\sigma}=\mathbf{1}}, \tag{5}$$

to approximate the LOO effect defined in (2). By applying the chain rule in (5), the key is to efficiently compute the influence scores of the data point z_{li} on the task-specific parameters $\hat{\theta}_k$ and shared parameters $\hat{\gamma}$.

To achieve this, we present the following proposition that provides the explicit analytical form for the influence of a data point on task-specific parameters for the same task (within-task influence), task-specific parameters for another task (between-task influence), and shared parameters (shared influence). Before introducing the results, we first define some notation. Let H_{kl} denote the (k, l)-th block components of the Hessian matrix of the MTL objective function $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$, as defined in (4), with respect to $\boldsymbol{w}$. This Hessian matrix has the following block structure:

$$H(\boldsymbol{w}, \boldsymbol{\sigma}) = \begin{pmatrix} H_{1,1} & \cdots & 0 & H_{1,K+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & H_{K,K} & H_{K,K+1} \\ H_{K+1,1} & \cdots & H_{K+1,K} & H_{K+1,K+1} \end{pmatrix}. \tag{6}$$

The details of each block are described in Lemma F.1. A naive computation of the influence function would require inverting the entire Hessian matrix in (6), with dimensions $(\sum_{k=1}^{K} d_k + p) \times (\sum_{k=1}^{K} d_k + p)$, which could be computationally expensive and numerically unstable. We take advantage of Hessian's block structure in MTL and simplify the computation to only require the inversion of submatrices.

Proposition 1 (Data-Level Within-task Influence, Between-task Influence, and Shared Influence)**.** *Assuming the objective function $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$ in (4) is twice-differentiable and strictly convex in $\boldsymbol{w}$. For any two tasks $k \neq l$ and $1 \leq k, l \leq K$, the following results hold:*

(Shared influence) For $1 \leq i \leq n_k$, the influence of the i-th data point from task k on the shared parameters, $\hat{\gamma}$, is given by

$$\frac{\partial \hat{\gamma}}{\partial \sigma_{ki}} = -\left(N^{-1} \cdot H_{K+1,k} H_{kk}^{-1} \frac{\partial \ell_{ki}}{\partial \theta_k} + N^{-1} \frac{\partial \ell_{ki}}{\partial \gamma} \right), \tag{7}$$

where the matrix $N := H_{K+1,K+1} - \sum_{k=1}^{K} H_{K+1,k} H_{kk}^{-1} H_{k,K+1} \in \mathbb{R}^{p \times p}$ is invertible;

(Within-task influence) For $1 \leq i \leq n_k$, the influence of the i-th data point from task k on the task-specific parameters for the same task k, $\hat{\theta}_k$, is given by

$$\frac{\partial \hat{\theta}_k}{\partial \sigma_{ki}} = -H_{kk}^{-1} \frac{\partial \ell_{ki}}{\partial \theta_k} + H_{kk}^{-1} H_{k,K+1} \cdot \frac{\partial \hat{\gamma}}{\partial \sigma_{ki}}; \tag{8}$$

(Between-task influence) For $1 \leq i \leq n_l$, the influence of the i-th data point from task l on the task-specific parameters for another task k, $\hat{\theta}_k$, is given by

$$\frac{\partial \hat{\theta}_k}{\partial \sigma_{li}} = H_{kk}^{-1} H_{k,K+1} \cdot \frac{\partial \hat{\gamma}}{\partial \sigma_{li}}. \tag{9}$$

Interpretation of Data-Level Influences In MTL, data points have more composite influences on task-specific parameters compared to STL due to interactions with other tasks and shared parameters. In STL, each data point only affects its own task's parameters through the gradient and Hessian of the task-specific objective, which is solely the first term in (8). However, in MTL, shared parameters introduce a feedback mechanism that allows data from one task to influence the parameters of other tasks. As shown in (7), the influence of i-th data point from task k on the shared parameters stem from two sources: the first term reflects the change on the task-specific parameter $\hat{\theta}_k$, which then indirectly affects the shared parameters $\hat{\gamma}$, while the second term accounts for the direct impact on $\hat{\gamma}$. Consequently, within-task influence in (8) includes an additional influence propagated through the shared parameters, and between-task influence in (9) arises as data from one task indirectly impacts the parameters of another task via the shared parameters. In particular, in STL, between-task influence does not occur because tasks are independent and do not interact.

3 Experiments

In this section, we empirically demonstrate the performance of MTIF in two experimental setups. In Section 3.1, we present results from a linear regression setup, as described in Example 1. In Section 3.2, we report results from a shared-bottom neural network setup, detailed in Example 2.

Through these experiments, we show the following benefits of MTIF. Firstly, our data-level influence score provides a strong approximation to the leave-one-out (LOO) effect in (2), as evidenced by the high degree of linearity between the two measures. Secondly, our task-level influence score effectively approximates the leave-one-task-out (LOTO) effect in (3), demonstrated by the strong correlation in task contribution rankings between the two measures. Moreover, data selection enabled by our method leads to both improved model performance across various dynamic weighting algorithms for MTL, and interpretable insights about the task relationships through case studies. Due to page limit, some results are shown in Appendix E.

3.1 Linear Regression

Our experiments in the linear regression setting, as described in Example 1, consist of evaluations on two datasets. The first dataset is a synthetic dataset with 10 tasks, each contains 200 samples (x_{ji}, y_{ji}) randomly split into training and testing set with equal size. The second dataset is a real-world dataset, HAR (Anguita et al., 2013), as referenced in Duan & Wang (2023). We leave more details of both datasets in Appendix H.1.

3.1.1 Data-level Influence

In this experiment, we compare our data-level influence scores (5) with the gold standard LOO scores (2) on both the synthetic and HAR datasets. We evaluate both within-in task influence and between-task influence.

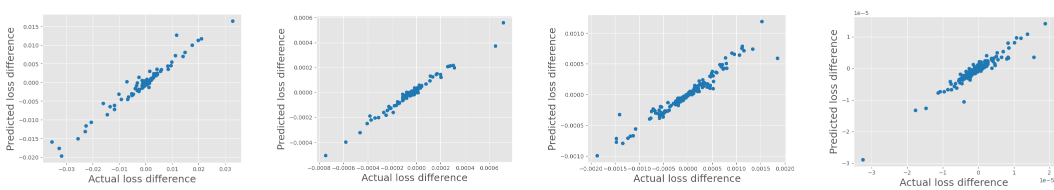

Figure 1: LOO experiments on linear regression. The x-axis is the actual loss difference obtained by LOO retraining, and the y-axis is the predicted loss difference calculated by MTIF. The first two figures from the left show within-task and between-task results (in order) results on the synthetic dataset, while the other two figures present within-task and between-task results (in order) on the HAR dataset. Each figure corresponds to a randomly picked test data point. The scatter points correspond to training data points in the first task of each dataset. The trend holds more broadly.

Figure 1 presents our results. The strong linear correlation between MTIF influence scores and the gold standard LOO scores across all scenarios indicates that MTIF effectively approximates the LOO effect, both for within-task and between-task influence, on both the synthetic and HAR datasets.

3.1.2 Task-level Influence

In this experiment, we compare our task-level influence scores (D.3) with the gold standard LOTO scores (3) on both the synthetic and HAR datasets. We randomly split 20% of the data from each task as the validation set. Specifically, for a given target task, we use MTIF to calculate the influence score of each training task to the model's performance on the target task's validation set. We also calculate the LOTO scores for each task by retraining the model. We then report the Spearman correlation coefficient between the MTIF influence scores and the LOTO scores. Table 1 shows the results on the synthetic dataset for each task selected as the target task. We leave the results for HAR dataset to Appendix H.1.2 due to space limit. On both datasets, the proposed MTIF achieves high correlation coefficients with the LOTO scores, indicating that MTIF aligns well with LOTO.

Table 1: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. Error bars indicate the standard error of the mean.

Task 1	Task 2	Task 3	Task 4	Task 5
0.84 ± 0.05	0.72 ± 0.05	0.74 ± 0.11	0.81 ± 0.05	0.71 ± 0.09
Task 6	Task 7	Task 8	Task 9	Task 10
0.74 ± 0.04	0.74 ± 0.07	0.84 ± 0.03	0.74 ± 0.03	0.65 ± 0.07

3.2 Neural Networks

We further demonstrate that the proposed MTIF remains effective for neural networks. We conduct the experiments with the Shared-Bottom neural network (as discussed in Example 2) on the CelebA dataset (Liu et al., 2015).

3.2.1 Application: Data Selection

We demonstrate the practical usefulness of the proposed MTIF through a downstream application of data selection. While most existing MTL literature predominantly investigates task relatedness at the task level, the data-level influence estimated by MTIF provides a unique opportunity to improve MTL by removing a small portion of training data points that cause negative impact to the overall performance.

In this experiment, we remove top 5% most negative training data points as estimated by MTIF on the validation dataset, and then report the test performance after retraining the MTL models on the rest of the training dataset.

As a reference, we include several re-weighting-based methods, such as GradNorm (GN) (Chen et al., 2018), Dynamic Weight Average (DWA) (Liu et al., 2019), Impartial MultiTask Learning (IMTL) (Liu et al., 2021a), Random Loss Weighting (RLW) (Lin et al., 2022), and Uncertainty Weighting (UW) (Kendall et al., 2018), which aim to improve MTL performance by dynamically measuring and accounting for the task relatedness during training. Since such re-weighting-based methods are orthogonal to data selection, we also experiment with combining re-weighting methods with data selection. We refer to the vanilla method without re-weighting as Equal Weighting (EW). We use EW+DS to represent the combination of Equal Weighting and data selection. We adopt the implementation from libMTL (Lin & Zhang, 2023) for all the re-weighting methods. Due to page limit, we only show the results for EW and EW+DS below in Table 2 (the full results are in Table 2 in Appendix E.1.2).

As shown in Table 2 (and full results in Table 4), we first observe that removing the most negative data points appears to be more effective than re-weighting methods. Among all the re-weighting methods, only RLW (0.889) outperforms the vanilla baseline EW (0.885) in terms of average performance, while the data selection EW+DS achieves an average performance of 0.892. Moreover, DS consistently leads to performance improvement when combining with different re-weighting methods. This result suggests that methods accounting for the fine-grained data-level influence may lead to better improvement for MTL compared to methods that only examine task-level relatedness.

Table 2: Results of model performance using different dynamic weighting methods, both with and without data selection (DS). The DS method removes the top 5% of the most negative data points based on the data-level influence scores estimated by MTIF. EW refers to Equal Weighting, which is the vanilla Shared-Bottom model without any re-weighting. The reported values are averaged over 5 random seeds, with † indicating standard error of the mean < 0.01 and * indicating standard error of the mean < 0.002. The last column shows the average performance across all tasks.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Average
EW	0.861†	0.813†	0.901†	0.784†	0.927†	0.946†	0.856†	0.918^{*}	0.957†	0.885^{*}
EW+DS	0.869†	0.818†	0.902†	0.792†	0.934†	0.952^{*}	0.868†	0.929†	0.958†	0.892^{*}

Finally, while it is a bit counter-intuitive that most re-weighting based methods fail to outperform the vanilla baseline EW, similar observations also present in the benchmark study by libMTL (Lin & Zhang, 2023).

4 Conclusion

In this work, we proposed the *MultiTask Influence Function* (**MTIF**), a novel data attribution method for multitask learning (MTL). MTIF efficiently estimates the influence of individual data points on task performance across multiple tasks, without the need for retraining. By leveraging the structure of MTL models, MTIF enables scalable and interpretable data-level and task-level influence analysis. Extensive experimental results demonstrate the effectiveness of the proposed methods.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. The information complexity of learning tasks, their structure and their distance. *Information and Inference: A Journal of the IMA*, 10(1):51–72, 01 2021. ISSN 2049-8772. doi: 10.1093/imaiai/iaaa033. URL `https://doi.org/10.1093/imaiai/iaaa033`.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3, 2013.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1899–1909. PMLR, 26–28 Aug 2020. URL `https://proceedings.mlr.press/v108/barshan20a.html`.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. doi: 10.1023/A:1007379606734. URL `https://doi.org/10.1023/A:1007379606734`.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/chen18a.html`.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2039–2050. Curran Associates, Inc., 2020.

Yaqi Duan and Kaizheng Wang. Adaptive and robust multi-task learning. *The Annals of Statistics*, 51(5):2015 – 2039, 2023. doi: 10.1214/23-AOS2319. URL `https://doi.org/10.1214/23-AOS2319`.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 845–850, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2139. URL `https://aclanthology.org/P15-2139`.

Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pp. 109–117, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138881. doi: 10.1145/1014052.1014067. URL `https://doi.org/10.1145/1014052.1014067`.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27503–27516. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/e77910ebb93b511588557806310f78f1-Paper.pdf`.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/ghorbani19c.html`.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying large language model generalization with influence functions, 2023. URL `https://arxiv.org/abs/2308.03296`.

Han Guo, Nazneen Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. FastIF: Scalable influence functions for efficient model interpretation and debugging. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10333–10350, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.808. URL `https://aclanthology.org/2021.emnlp-main.808`.

Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: a survey. *Machine Learning*, 113(5):2351–2403, 2024.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1923–1933, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1206. URL `https://aclanthology.org/D17-1206`.

Zhen Huang, Jinyu Li, Sabato Marco Siniscalchi, I-Fan Chen, Ji Wu, and Chin-Hui Lee. Rapid adaptation for deep neural networks through multi-task learning. In *Interspeech 2015*, pp. 3625–3629, 2015. doi: 10.21437/Interspeech.2015-719.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9525–9587. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/ilyas22a.html`.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1167–1176. PMLR, 16–18 Apr 2019. URL `https://proceedings.mlr.press/v89/jia19a.html`.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL `https://arxiv.org/abs/1412.6980`.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/koh17a.html`.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/kwon22a.html.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9m02ib92Wz.

Dongyue Li, Huy Nguyen, and Hongyang Ryan Zhang. Identification of negative transfers in multitask learning using surrogate models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=KgfFAI9f3E. Featured Certification.

Baijiong Lin and Yu Zhang. Libmtl: A python library for deep multi-task learning. *The Journal of Machine Learning Research*, 24(1):9999–10005, 2023.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jjtFD8A1Wx.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=IMPnRXEWpvr.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Shiming Liu, Yifan Xia, Zhusheng Shi, Hui Yu, Zhiqiang Li, and Jianguo Lin. Deep learning in sheet metal bending with a novel theory-guided deep neural network. *IEEE/CAA Journal of Automatica Sinica*, 8(3):565–581, 2021b. doi: 10.1109/JAS.2021.1003871.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1930–1939, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220007. URL https://doi.org/10.1145/3219819.3220007.

Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning, 2016. URL https://arxiv.org/abs/1511.06342.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Xinyu Peng, Cheng Chang, Fei-Yue Wang, and Li Li. Robust multitask learning with sample gradient similarity. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(1):497–506, 2024. doi: 10.1109/TSMC.2023.3315541.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020.

Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017. URL https://arxiv.org/abs/1706.05098.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation, 2016. URL https://arxiv.org/abs/1511.06295.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8179–8186, Jun. 2022.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9120–9132. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/standley20a.html.

Jiachen T. Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6388–6421. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/wang23e.html.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4438–4450, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.359. URL https://aclanthology.org/2020.emnlp-main.359.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=F1vEjWK-lH_.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylzhkBtDB.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(2):35–63, 2007. URL http://jmlr.org/papers/v8/xue07a.html.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Wujie Zhou, Shaohua Dong, Jingsheng Lei, and Lu Yu. Mtanet: Multitask-aware network with hierarchical multimodal fusion for rgb-t urban scene understanding. *IEEE Transactions on Intelligent Vehicles*, 8(1):48–58, 2023. doi: 10.1109/TIV.2022.3164899.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76, 2021. doi: 10.1109/JPROC.2020.3004555.

A Related Work

Data Attribution Data attribution methods quantify the influence of individual training data points on model performance. These methods can be broadly categorized into retraining-based and gradient-based approaches (Hammoudeh & Lowd, 2024). Retraining-based methods (Ghorbani & Zou, 2019; Jia et al., 2019; Kwon & Zou, 2022; Wang & Jia, 2023; Ilyas et al., 2022) require retraining the model multiple times on different subsets of the training data. Retraining-based methods are usually computationally expensive due to the repeated retraining. The computation cost can be further exacerbated in MTL due to the combination of tasks. Gradient-based methods (Koh & Liang, 2017; Guo et al., 2021; Barshan et al., 2020; Schioppa et al., 2022; Kwon et al., 2024; Yeh et al., 2018; Pruthi et al., 2020; Park et al., 2023) instead rely on the (higher-order) gradient information of the original model to estimate the data influence, which are more efficient. Many gradient-based methods can be viewed as variants of IF-based data attribution methods (Koh & Liang, 2017). In this paper, we develop an IF-based data attribution method tailored for the MTL settings.

Task Relatedness in Multitask Learning Quantifying task relatedness has been a central focus in multitask learning. Broadly, two lines of work address this topic. The first focuses on task grouping or task selection, aiming to develop methods for grouping or selecting positively related tasks to improve prediction performance. Standley et al. (2020) and Li et al. (2023) introduced task selection methods based on model retraining, which are less efficient than our method. Fifty et al. (2021) proposed an efficient method for calculating heuristic pairwise task affinities, but their estimator heavily depends on the training trajectory, which limits its interpretability. Additionally, Wu et al. (2020) incorporated task data covariance to estimate task similarity, though their work is restricted to specific types of models.

Another line of research focuses on developing advanced training algorithms for MTL by explicitly accounting for inter-task relations during training. These methods generally fall into two categories. The first category manipulates per-task gradients to mitigate negative influences between tasks (Yu et al., 2020; Wang et al., 2021; Liu et al., 2021a; Chen et al., 2020; Peng et al., 2024). The second category employs task reweighting techniques to balance the contribution of each task or to emphasize on critical tasks (Chen et al., 2018; Liu et al., 2019; Guo et al., 2018; Kendall et al., 2018). Beyond these two categories, Duan & Wang (2023) proposed a family of methods that automatically leverage task similarities to improve multitask learning. These approaches are orthogonal to our method and can be potentially combined with the data and task selection enabled by our method.

There is also a body of work on task relatedness in transfer learning (Zamir et al., 2018; Achille et al., 2021; Dwivedi & Roig, 2019; Zhuang et al., 2021; Achille et al., 2019). However, Standley et al. (2020) demonstrated that task similarity metrics in transfer learning do not generalize well to the multitask learning domain.

B Preliminary: Influence Function as an Approximation to LOO

As a widely used data attribution metric, the leave-one-out (LOO) effect measures the contribution of a training data point by the change of model performance after removing this data point and retraining the model (Koh & Liang, 2017; Schioppa et al., 2022; Grosse et al., 2023). However, repeatedly retraining the model can be computationally extensive. To address this issue, in the single-task learning (STL) setting, Koh & Liang (2017) proposed the use of influence functions, which approximate the LOO effect by leveraging small perturbations to the weight of the loss at each data point.

Specifically, for a given data point $z \in \mathcal{Z}$ and parameter vector $\theta \in \Theta$, consider a loss function $\ell(\theta; z)$. Given a training dataset $\{z_i\}_{i=1}^n$, we minimize the empirical risk, i.e., $\hat{\theta} = \arg\min_{\theta\in\Theta} \sum_{i=1}^n \ell(\theta; z_i)/n$, and evaluate the performance of $\hat{\theta}$ using certain evaluation metrics. A common metric is the average loss on the validation data D^{v}, i.e., $V(\hat{\theta}; D^{\text{v}}) = \sum_{z\in D^{\text{v}}} \ell(\hat{\theta}; z)/|D^{\text{v}}|$. The LOO effect of the i-th data point is defined as the difference in the evaluation metric when using the parameters learned from all data points versus the parameters learned by excluding the data point z_i. Formally, we introduce a weight vector $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_n)$ into the objective function, then the minimizer can be written by

$$\hat{\theta}(\boldsymbol{\sigma}) = \arg\min_{\theta\in\Theta} \mathcal{L}(\theta, \boldsymbol{\sigma}), \text{ where } \mathcal{L}(\theta, \boldsymbol{\sigma}) := \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(\theta; z_i).$$

The LOO effect of the i-th data point is given by $V(\hat{\theta}(\mathbf{1}); D^{\mathrm{v}}) - V(\hat{\theta}(\mathbf{1}^{(-i)}); D^{\mathrm{v}})$, where $\mathbf{1}$ is an all-ones vector with length n and $\mathbf{1}^{(-i)}$ is a vector of all ones except for the i-th element being 0. The LOO effect requires retraining the model multiple times — once for each data point being left out - to obtain $\hat{\theta}(\mathbf{1}^{(-i)})$. To reduce computational cost, Koh & Liang (2017) proposed to approximate the LOO effect by using the partial derivative $\nabla_\theta V(\hat{\theta}(\boldsymbol{\sigma}); D^{\mathrm{v}})^\top \cdot \frac{\partial \hat{\theta}(\boldsymbol{\sigma})}{\partial \sigma_i}\Big|_{\boldsymbol{\sigma}=\mathbf{1}}$. Under certain regularity conditions, the effect of perturbing the weight for data point z_i on the learned parameters is given by

$$\frac{\partial \hat{\theta}(\boldsymbol{\sigma})}{\partial \sigma_i}\Big|_{\boldsymbol{\sigma}=\mathbf{1}} = -H(\hat{\theta}(\mathbf{1}), \mathbf{1})^{-1} \cdot \nabla_\theta \ell(\hat{\theta}(\mathbf{1}); z_i), \tag{B.1}$$

where $H(\theta, \boldsymbol{\sigma}) = \sum_{i=1}^{n} \sigma_i \frac{\partial^2 \ell(\theta; z_i)}{\partial\theta\partial\theta^\top}/n$ is the Hessian matrix. This approximation is referred to as influence function (IF)-based data attribution. Compared to the LOO effect, IF-based data attribution only requires the evaluation of the inverse Hessian matrix and the gradient at the model parameters trained on the full dataset.

While IF-based data attribution has been shown as a scalable and effective tool for many applications, it has been primarily developed for STL settings, where a single model is trained on a homogeneous task. However, in many real-world applications, multiple related tasks are learned jointly, with shared parameters across tasks and different evaluation metrics of interest. In the next section, we extend IF-based data attribution to the multitask learning (MTL) setting, broadening its applicability.

C Examples of MTL Models

Example 1 (Multitask Linear Regression with Ridge Penalty). *Regularization has been integrated in MTL to encourange similarity among task-specific parameters; see (Evgeniou & Pontil, 2004; Duan & Wang, 2023) for examples. Consider the regression setting where $y_{ki} = x_{ki}^\top \theta_k^* + \epsilon_{ki}$, with ϵ_{ki} being independent noise and $x_{ki} \in \mathbb{R}^d$ for $1 \le i \le n_k$ and $1 \le k \le K$. Additionally, we have the prior knowledge that $\{\theta_k^*\}_{k=1}^K$ are close to each other. Instead of fitting a separate ordinary least squares estimator for each θ_k, a ridge penalty is introduced to shrink the task-specific parameters $\theta_1, \ldots, \theta_K \in \mathbb{R}^d$ toward a common vector $\gamma \in \mathbb{R}^d$, while γ is simultaneously learned by leveraging data from all tasks.*

The objective function for multitask linear regression with a ridge penalty is given by

$$\mathcal{L}(\boldsymbol{w}) = \sum_{k=1}^{K} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} (y_{ki} - x_{ki}^\top \theta_k)^2 + \lambda_k \|\theta_k - \gamma\|_2^2 \right],$$

where λ_k controls the strength of regularization. This can be viewed as a special case of (1) by setting ℓ_k as the squared error (depending only on the task-specific parameters) and defining the regularization term $\Omega_k(\theta_k, \gamma) = \lambda_k \|\theta_k - \gamma\|_2^2$.

Example 2 (Shared-Bottom Neural Network Model). *The shared-bottom neural network architecture, first proposed by Caruana (1997), has been widely applied to MTL across various domains (Zhou et al., 2023; Liu et al., 2021b; Ma et al., 2018). The shared-bottom model can be represented as $f_k(x) = g(\theta_k; f(\gamma; x))$, where $f(\gamma; \cdot)$ represents the shared layers that process the input data and produce an intermediate representation, and γ denotes the parameters shared across tasks. The function $g(\theta_k; \cdot)$ corresponds to task-specific layers, which take the intermediate representation and produce task-specific predictions, with θ_k representing task-specific parameters.*

The loss function for this model can be written as:

$$\mathcal{L}(\boldsymbol{w}) = \sum_{k=1}^{K} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \ell_k(y_{ki}, g(\theta_k; f(\gamma; x_{ki}))) + \Omega_k(\theta_k, \gamma) \right],$$

where $\ell_k(\cdot, \cdot)$ represents the task-specific loss function, and $\Omega_k(\theta_k, \gamma)$ denotes the regularization term. A simple choice for regularization is $\Omega_k(\theta_k, \gamma) = \lambda_k(\|\theta_k\|_2^2 + c\|\gamma\|_2^2)$, where λ_k and c are positive constants.

D Task-Level Influences

The LOTO effect, introduced in (3) is a natural measure for task relatedness. To provide a computationally efficient approximation of the LOTO effect, we similarly apply infinitesimal perturbations on the data weights. Specifically, we consider the following task-level $\boldsymbol{\sigma}$-weighted objective, where we assign the same weight to data from the same task:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma}) = \sum_{k=1}^{K} \sigma_k \left[\frac{1}{n_k} \sum_{i=1}^{n_k} \ell_{ki}(\theta_k, \gamma) + \Omega_k(\theta_k, \gamma) \right]. \tag{D.2}$$

Note that, the regularization terms $\Omega_k(\theta_k, \gamma)$ in (D.2) are also weighted by σ_k, unlike the data-level $\boldsymbol{\sigma}$-weighted objective (4), where the weights are only applied to the individual losses $l_{ki}(\theta_k, \gamma)$. This difference is due to the nature of multitask learning - excluding a task results in the removal of its task-specific parameters along with the regularization term.

The IF-based approximation for the LOTO effect Δ_k^l is given by

$$\frac{\partial V_k(\hat{\theta}_k(\boldsymbol{\sigma}), \hat{\gamma}(\boldsymbol{\sigma}); D_k^{\mathrm{v}})}{\partial \sigma_l}\bigg|_{\boldsymbol{\sigma}=\mathbf{1}} = \nabla_\theta V_k(\hat{\theta}_k, \hat{\gamma}; D_k^{\mathrm{v}}) \cdot \frac{\partial \hat{\theta}_k(\boldsymbol{\sigma})}{\partial \sigma_l}\Big|_{\boldsymbol{\sigma}=\mathbf{1}} + \nabla_\gamma V_k(\hat{\theta}_k, \hat{\gamma}; D_k^{\mathrm{v}}) \cdot \frac{\partial \hat{\gamma}(\boldsymbol{\sigma})}{\partial \sigma_l}\Big|_{\boldsymbol{\sigma}=\mathbf{1}}. \tag{D.3}$$

In Proposition 2, we provide the analytical form for the influence of data from one task on the parameters of another task and the shared parameters. The Hessian matrix of $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$ with respect to $\boldsymbol{w}$ shares the same block structure as shown in (6). Let H_{kl} denote the (k, l)-th block of the Hessian matrix, with the details provided in Lemma F.2. Let N be defined as in Proposition 1.

Proposition 2 (Task-Level Between-task Influence)**.** *Under the assumptions of Proposition 1, for any two tasks $k \neq l$ where $1 \leq k, l \leq K$, the influence of data from task l on the task-specific parameters of task k, $\hat{\theta}_k$, is given by*

$$\frac{\partial \hat{\theta}_k}{\partial \sigma_l} = H_{kk}^{-1} H_{k,K+1} \cdot \frac{\partial \hat{\gamma}}{\partial \sigma_l}, \tag{D.4}$$

where $\frac{\partial \hat{\gamma}}{\partial \sigma_l}$ is the influence of data from task l on the shared parameters, $\hat{\gamma}$, and is given by

$$\frac{\partial \hat{\gamma}}{\partial \sigma_l} = -\left(N^{-1} H_{K+1,l} H_{ll}^{-1} \left[\sum_{i=1}^{n_l} \frac{\partial \ell_{li}}{\partial \theta_l} + \frac{\partial \Omega_l}{\partial \theta_l} \right] + N^{-1} \left[\sum_{i=1}^{n_l} \frac{\partial \ell_{li}}{\partial \gamma} + \frac{\partial \Omega_l}{\partial \gamma} \right] \right). \tag{D.5}$$

Proof. The result follows directly from the application of Lemma F.4 and Lemma F.6. □

As shown in Proposition 2, task-level influences $\frac{\partial \hat{\theta}_k}{\partial \sigma_l}$ and $\frac{\partial \hat{\gamma}}{\partial \sigma_l}$ are sums of data-level influence scores for all points in task l, with additional terms arising from σ-weighted regularization.

E More Experiments

E.1 Neural Networks

E.1.1 Task-level Influence

In this experiment, we compare the task-level influence estimated by MTIF with the gold standard LOTO scores on the neural network setting, following a similar setup as the linear model setting in Sec 3.1.2. Table 3 reports the average Spearman correlation coefficients across 5 random seeds with each task selected as the target task. In comparison to the linear model setting in Table 1, the correlation coefficients are lower. This is not surprising as data attribution for non-convex models is more challenging and the evaluation is more noisy due to the inherent randomness in model retraining (Koh & Liang, 2017). Nevertheless, the influence scores estimated by MTIF still demonstrate non-trivial correlations with the LOTO scores in most cases, with the highest correlation coefficient achieving 0.43. This indicates that MTIF still effectively captures useful signals.

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9
0.21 ± 0.04	0.35 ± 0.19	0.23 ± 0.10	0.43 ± 0.12	0.14 ± 0.17	0.36 ± 0.10	0.29 ± 0.05	0.10 ± 0.10	0.15 ± 0.15

Table 3: The average Spearman correlation coefficients over 5 random seeds on the CelebA dataset.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Average
DWA	0.864†	0.804†	0.903†	0.786†	0.931†	0.945*	0.854†	0.920*	0.955†	0.885*
DWA+DS	0.873†	0.815†	0.904†	0.795†	0.937†	0.950*	0.866†	0.929*	0.956†	0.892*
GN	0.864†	0.808†	0.900†	0.781†	0.928†	0.946†	0.856†	0.922†	0.954†	0.884*
GN+DS	0.873†	0.817†	0.898†	0.791†	0.934†	0.951*	0.870†	0.931*	0.957†	0.891*
IMTL	0.858†	0.800†	0.896†	0.775†	0.930†	0.947*	0.869†	0.917†	0.957†	0.883*
IMTL+DS	0.871*	0.806†	0.897†	0.788†	0.934†	0.952*	0.873†	0.925†	0.961†	0.890*
UW	0.857†	0.808†	0.897†	0.781†	0.925†	0.945*	0.859†	0.920*	0.956†	0.883*
UW+DS	0.867†	0.816†	0.900†	0.792†	0.931†	0.953*	0.866†	0.929†	0.958†	0.890*
RLW	0.870†	0.821†	0.901†	0.789†	0.934†	0.951†	0.856†	0.927*	0.955†	0.889*
RLW+DS	0.881*	0.820†	0.907†	0.798†	0.936†	0.954*	0.868*	0.932*	0.958†	0.895*
EW	0.861†	0.813†	0.901†	0.784†	0.927†	0.946†	0.856†	0.918*	0.957†	0.885*
EW+DS	0.869†	0.818†	0.902†	0.792†	0.934†	0.952*	0.868†	0.929†	0.958†	0.892*

Table 4: Results of model performance using different dynamic weighting methods, both with and without data selection (DS). The DS method removes the top 5% of the most negative data points based on the data-level influence scores estimated by MTIF. EW refers to Equal Weighting, which is the vanilla Shared-Bottom model without any re-weighting. The reported values are averaged over 5 random seeds, with † indicating standard error of the mean < 0.01 and * indicating standard error of the mean < 0.002. The last column shows the average performance across all tasks.

E.1.2 Full Results of Data Selection

E.1.3 Visualization of Most Negative Samples

Finally, we demonstrate how MTIF might bring us interpretable insights about task relatedness. In Figure 2, we visualize some of the most negative samples between specific task pairs.

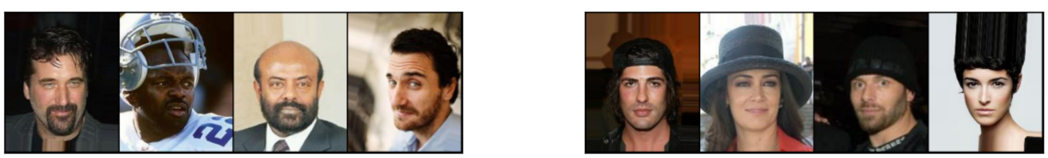

Figure 2: The images on the left represent four samples from the task "Mustache" that negatively influence the task "No Beard." They are labeled positive for "Mustache" but negative for "No Beard." On the right, there are four samples from the task "Wearing Hat" that negatively influence the task "Black Hair." They are labeled positive for "Wearing Hat" but negative for "Black Hair."

On the left side of Figure 2, we visualize the samples from the task "Mustache" that negatively influence the task "No Beard". Intuitively, these two tasks are related tasks, as someone with a mustache is certainly with a beard. However, these images are all negative samples for the task "Mustache," yet the individuals clearly have beards. These images could potentially confuse the model. Similarly, the images on the right depict individuals all wearing a black hat, but are labeled as not having black hair, either because their hair is not visible in the picture or because their natural hair color is not black, though this is not obvious from the image. The model may confuse the presence of a black hat with having black hair. These examples show that MTIF is capable of finding samples from one task that negatively influence another task, offering interpretable insights about task relationships.

F Lemma

The first two lemmas describe the structure of the Hessian matrices for data-level and task-level inference.

Lemma F.1 (Hessian Matrix Structure for Data-Level Inference)**.** *Let $H(\boldsymbol{w}, \boldsymbol{\sigma})$ be the Hessian matrix of data-level $\boldsymbol{\sigma}$-weighted objective (4) with respect to $\boldsymbol{w}$, i.e., $H(\boldsymbol{w}, \boldsymbol{\sigma}) = \frac{\partial^2 \mathcal{L}(\boldsymbol{w},\boldsymbol{\sigma})}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\top}$, then we have*

$$H(\boldsymbol{w}, \boldsymbol{\sigma}) = \begin{pmatrix} H_{1,1} & \cdots & 0 & H_{1,K+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & H_{K,K} & H_{K,K+1} \\ H_{K+1,1} & \cdots & H_{K+1,K} & H_{K+1,K+1} \end{pmatrix},$$

where

$$\begin{aligned} H_{kk} &= \sum_{i=1}^{n_k} \sigma_{ki} \frac{\partial^2 \ell_{ki}(\theta_k, \gamma)}{\partial \theta_k \partial \theta_k^\top} + \frac{\partial^2 \Omega_k(\theta_k, \gamma)}{\partial \theta_k \partial \theta_k^\top} \in \mathbb{R}^{d_k \times d_k} \quad \textit{for } 1 \le k \le K, \\ H_{kl} &= 0 \in \mathbb{R}^{d_k \times d_l} \quad \textit{for } 1 \le k, l \le K \textit{ and } k \ne l, \\ H_{K+1,k}^\top = H_{k,K+1} &= \sum_{i=1}^{n_k} \sigma_{ki} \frac{\partial^2 \ell_{ki}(\theta_k, \gamma)}{\partial \theta_k \partial \gamma^\top} + \frac{\partial^2 \Omega_k(\theta_k, \gamma)}{\partial \theta_k \partial \gamma^\top} \in \mathbb{R}^{d_k \times p} \quad \textit{for } 1 \le k \le K, \\ H_{K+1,K+1} &= \sum_{k=1}^{K} \sum_{i=1}^{n_k} \sigma_{ki} \frac{\partial^2 \ell_{ki}(\theta_k, \gamma)}{\partial \gamma \partial \gamma^\top} + \sum_{k=1}^{K} \frac{\partial^2 \Omega_k(\theta_k, \gamma)}{\partial \gamma \partial \gamma^\top} \in \mathbb{R}^{p \times p}. \end{aligned}$$

Lemma F.2 (Hessian Matrix Structure for Task-Level Inference)**.** *Let $H(\boldsymbol{w}, \boldsymbol{\sigma})$ be the Hessian matrix of task-level $\boldsymbol{\sigma}$-weighted objective (D.2) with respect to $\boldsymbol{w}$, then*

$$H(\boldsymbol{w}, \boldsymbol{\sigma}) = \begin{pmatrix} H_{1,1} & \cdots & 0 & H_{1,K+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & H_{K,K} & H_{K,K+1} \\ H_{K+1,1} & \cdots & H_{K+1,K} & H_{K+1,K+1} \end{pmatrix},$$

where

$$\begin{aligned} H_{kk} &= \sigma_k \left[\sum_{i=1}^{n_k} \frac{\partial^2 \ell_{ki}(\theta_k, \gamma)}{\partial \theta_k \partial \theta_k^\top} + \frac{\partial^2 \Omega_k(\theta_k, \gamma)}{\partial \theta_k \partial \theta_k^\top} \right] \in \mathbb{R}^{d_k \times d_k} \quad \textit{for } 1 \le k \le K, \\ H_{kl} &= 0 \in \mathbb{R}^{d_k \times d_l} \quad \textit{for } 1 \le k, l \le K \textit{ and } k \ne l, \\ H_{K+1,k}^\top = H_{k,K+1} &= \sigma_k \left[\sum_{i=1}^{n_k} \frac{\partial^2 \ell_{ki}(\theta_k, \gamma)}{\partial \theta_k \partial \gamma^\top} + \frac{\partial^2 \Omega_k(\theta_k, \gamma)}{\partial \theta_k \partial \gamma^\top} \right] \in \mathbb{R}^{d_k \times p} \quad \textit{for } 1 \le k \le K, \\ H_{K+1,K+1} &= \sum_{k=1}^{K} \sigma_k \left[\sum_{i=1}^{n_k} \frac{\partial^2 \ell_{ki}(\theta_k, \gamma)}{\partial \gamma \partial \gamma^\top} + \frac{\partial^2 \Omega_k(\theta_k, \gamma)}{\partial \gamma \partial \gamma^\top} \right] \in \mathbb{R}^{p \times p}. \end{aligned}$$

Lemma F.3 (Influence Scores for Data-Level Analysis)**.** *Assume that the objective $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$ is twice differentiable and strictly convex in $\boldsymbol{w}$. Then, $\hat{\boldsymbol{w}}(\boldsymbol{\sigma}) = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$ satisfies $\frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}(\boldsymbol{\sigma}), \boldsymbol{\sigma})}{\partial \boldsymbol{w}} = 0$. Moreover, we have:*

$$\frac{\partial \hat{\boldsymbol{w}}(\boldsymbol{\sigma})}{\partial \sigma_{ki}} = -H(\hat{\boldsymbol{w}}(\boldsymbol{\sigma}), \boldsymbol{\sigma})^{-1} \left(0, \cdots, 0, \underbrace{\frac{\partial \ell_{ki}}{\partial \theta_k^\top}}_{k\text{-th block}}, 0, \cdots, 0, \underbrace{\frac{\partial \ell_{ki}}{\partial \gamma^\top}}_{(K+1)\text{-th block}} \right)^\top,$$

where $H(\boldsymbol{w}, \boldsymbol{\sigma}) \in \mathbb{R}^{(\sum_{k=1}^K d_k + p) \times (\sum_{k=1}^K d_k + p)}$ is the Hessian matrix of $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\sigma})$ with respect to $\boldsymbol{w}$.

Proof. The result is obtained by applying the classical influence function framework as outlined in Koh & Liang (2017). □

Lemma F.4 (Influence Scores for Task-Level Analysis)**.** *Assume that the objective $\mathcal{L}(\boldsymbol{w},\boldsymbol{\sigma})$ is twice differentiable and strictly convex in $\boldsymbol{w}$. Then, the optimal solution $\hat{\boldsymbol{w}}(\boldsymbol{\sigma}) = \arg\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w},\boldsymbol{\sigma})$ satisfies $\frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}(\boldsymbol{\sigma}),\boldsymbol{\sigma})}{\partial \boldsymbol{w}} = 0$. Furthermore, we have:*

$$\frac{\partial \hat{\boldsymbol{w}}(\boldsymbol{\sigma})}{\partial \sigma_k} = -H(\hat{\boldsymbol{w}}(\boldsymbol{\sigma}),\boldsymbol{\sigma})^{-1}\left(0,\cdots,0,\underbrace{\sum_{i=1}^{n_k}\frac{\partial \ell_{ki}}{\partial \theta_k}+\frac{\partial \Omega_k}{\partial \theta_k}}_{k\text{-th block}},0,\cdots,0,\underbrace{\sum_{i=1}^{n_k}\frac{\partial \ell_{ki}}{\partial \gamma}+\frac{\partial \Omega_k}{\partial \gamma}}_{(K+1)\text{-th block}}\right)^{\top},$$

where $H(\boldsymbol{w},\boldsymbol{\sigma}) \in \mathbb{R}^{(\sum_{k=1}^K d_k+p)\times(\sum_{k=1}^K d_k+p)}$ is the Hessian matrix of $\mathcal{L}(\boldsymbol{w},\boldsymbol{\sigma})$ with respect to $\boldsymbol{w}$.

Proof. The result is obtained by applying the classical influence function framework as outlined in Koh & Liang (2017). □

The following two lemmas provide tools for verifying the invertibility of the Hessian matrix and calculating its inverse.

Lemma F.5 (Invertibility of Hessian)**.** *If H_{kk} is invertible for $1 \le k \le K$, define*

$$N := H_{K+1,K+1} - \sum_{k=1}^{K} H_{K+1,k}H_{kk}^{-1}H_{k,K+1} \in \mathbb{R}^{p\times p}. \tag{F.6}$$

If N is also invertible, then H is invertible.

Proof. The proof is in Section G. □

Lemma F.6 (Hessian Inverse)**.** *Let $\left[H^{-1}\right]_{k,l}$ denote the (k,l) block of the inverse Hessian $H(\boldsymbol{w},\boldsymbol{\sigma})^{-1}$. Then for $1 \le k,l \le K$,*

$$\begin{aligned}
\left[H^{-1}\right]_{k,l} &= \mathbf{1}(k=l)\cdot H_{kk}^{-1} + H_{kk}^{-1}H_{k,K+1}N^{-1}H_{K+1,l}H_{ll}^{-1}, && \textit{for } 1\le k,l\le K,\\
\left[H^{-1}\right]_{k,K+1} &= H_{kk}^{-1}H_{k,K+1}N^{-1}, && \textit{for } 1\le k\le K,\\
\left[H^{-1}\right]_{K+1,K+1} &= N^{-1}.
\end{aligned}$$

Proof. The proof is in Section G. □

G Proof

Proof of Lemma F.5 and Lemma F.6. Denote

$$H = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where $A = \begin{pmatrix} H_{11} & & 0 \\ & \ddots & \\ 0 & & H_{KK}\end{pmatrix} \in \mathbb{R}^{(\sum_{k=1}^K n_k)\times(\sum_{k=1}^K n_k)}$, $B = C^{\top} = \begin{pmatrix} H_{1,K+1} \\ \vdots \\ H_{K,K+1}\end{pmatrix} \in \mathbb{R}^{(\sum_{k=1}^K n_k)\times p}$, and $D = H_{K+1,K+1} \in \mathbb{R}^{p\times p}$. Under the conditions, the matrices H_{kk} for $1 \le k \le K$ are invertible. Note that A is a diagonal block matrix. It is also invertible and its inverse is given by

$$A^{-1} = \begin{pmatrix} H_{11}^{-1} & & \\ & \ddots & \\ & & H_{KK}^{-1}\end{pmatrix}.$$

In addition, under the conditions, $D - CA^{-1}B = H_{K+1,K+1} - \sum_{k=1}^{K} H_{K+1,k}H_{kk}^{-1}H_{k,K+1} = N$ is invertible. Using the inverse formula for block matrix, we have

$$H^{-1} = \begin{pmatrix} \left(A - BD^{-1}C\right)^{-1} & -A^{-1}B\left(D - CA^{-1}B\right)^{-1} \\ -D^{-1}C\left(A - BD^{-1}C\right)^{-1} & \left(D - CA^{-1}B\right)^{-1}\end{pmatrix}, \tag{G.7}$$

where the upper left block is equivalent to

$$\left(A - BD^{-1}C\right)^{-1} = A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1}CA^{-1},$$

by using the Woodbury matrix identity. Further, by expanding the RHS of Equation (G.7) in terms of the blocks in H, we can get the block-wise expression of H^{-1}. In particular, for $1 \le k, l \le K$,

$$\begin{aligned}\left[H^{-1}\right]_{k,l} \equiv \left[\left(A - BD^{-1}C\right)^{-1}\right]_{k,l} &= 1(k = l) \cdot H_{kk}^{-1} + \left[A^{-1}B\left(D - CA^{-1}B\right)^{-1}CA^{-1}\right]_{kl} \\ &= 1(k = l) \cdot H_{kk}^{-1} + H_{kk}^{-1}H_{k,K+1} \cdot N^{-1} \cdot H_{K+1,l}H_{ll}^{-1}.\end{aligned}$$

Further, for $1 \le k \le K$,

$$\left[H^{-1}\right]_{k,K+1} = \left[H^{-1}\right]_{K+1,k}^{\top} = H_{kk}^{-1}H_{k,K+1}N^{-1},$$

and

$$\left[H^{-1}\right]_{K+1,K+1} = N^{-1}.$$

□

H Experiment Details

H.1 Detailed Description of Datasets

H.1.1 Synthetic Dataset

The synthetic data for multi-task linear regression is generated with $m = 10$ tasks, where each dataset contains $n = 200$ samples (x_{ji}, y_{ji}) split into training set and test set. The input vectors x_{ji} are sampled independently from a normal distribution $\mathcal{N}(0, I_d)$ with dimensionality $d = 50$. The response y_{ji} is generated using a linear model $y_{ji} = x_{ji}^{\top}\theta_j^{\star} + \epsilon_{ji}$, where $\epsilon_{ji} \sim \mathcal{N}(0, 1)$ is independent noise.

The coefficient vectors $\theta_j^{\star}$ for task j are generated by setting a common vector $\beta^{\star} = 2e_1$ and adding random perturbations δ_j with norm δ sampled from a sphere. For a fraction ϵm of the tasks, the corresponding $\theta_j^{\star}$ are replaced with i.i.d. random vectors. Different methods, such as vanilla ARMUL (Duan & Wang, 2023) and independent task learning, are compared against this data generation.

Therefore, δ and ϵ are two parameters controlling the task similarity. The higher δ or ϵ is, the more dissimilar the tasks will be likely to be synthesized. We refer the readers for more detailed illustration in Duan & Wang (2023). Here, we provide several other results with different δ and ϵ.

Leave One Task Out (LOTO) From Table 5 Table 6 Table 7 Table 8 Table 9 Table 10, it is evident that our attribution score continues to perform strongly across various values of δ and ϵ. Notably, as δ and ϵ increase, the alignment between our task influence measure and the ground truth improves. This corresponds to the fact that tasks become more dissimilar, leading to a greater influence of each task on others due to the shared parameter γ. As a result, our task influence measure is better able to approximate the true relationships between tasks.

Task 1	Task 2	Task 3	Task 4	Task 5
0.84 ± 0.05	0.72 ± 0.05	0.74 ± 0.11	0.81 ± 0.05	0.71 ± 0.09
Task 6	Task 7	Task 8	Task 9	Task 10
0.74 ± 0.04	0.74 ± 0.07	0.84 ± 0.03	0.74 ± 0.03	0.65 ± 0.07

Table 5: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. $\delta = 1.0$ and $\epsilon = 0.2$

Task 1	Task 2	Task 3	Task 4	Task 5
0.75 ± 0.07	0.67 ± 0.06	0.81 ± 0.03	0.70 ± 0.05	0.60 ± 0.10
Task 6	Task 7	Task 8	Task 9	Task 10
0.39 ± 0.13	0.66 ± 0.06	0.75 ± 0.03	0.71 ± 0.05	0.61 ± 0.03

Table 6: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. $\delta = 1.0$ and $\epsilon = 0$.

Task 1	Task 2	Task 3	Task 4	Task 5
0.84 ± 0.04	0.67 ± 0.07	0.69 ± 0.12	0.77 ± 0.05	0.71 ± 0.05
Task 6	Task 7	Task 8	Task 9	Task 10
0.73 ± 0.07	0.65 ± 0.06	0.77 ± 0.05	0.69 ± 0.05	0.56 ± 0.11

Table 7: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. $\delta = 0.6$ and $\epsilon = 0.2$.

Task 1	Task 2	Task 3	Task 4	Task 5
0.77 ± 0.05	0.56 ± 0.09	0.69 ± 0.07	0.63 ± 0.06	0.57 ± 0.13
Task 6	Task 7	Task 8	Task 9	Task 10
0.38 ± 0.16	0.62 ± 0.04	0.72 ± 0.03	0.65 ± 0.04	0.46 ± 0.09

Table 8: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. $\delta = 0.6$ and $\epsilon = 0$.

Task 1	Task 2	Task 3	Task 4	Task 5
0.79 ± 0.05	0.62 ± 0.06	0.56 ± 0.13	0.73 ± 0.05	0.64 ± 0.07
Task 6	Task 7	Task 8	Task 9	Task 10
0.67 ± 0.08	0.52 ± 0.05	0.70 ± 0.04	0.65 ± 0.04	0.56 ± 0.09

Table 9: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. $\delta = 0.4$ and $\epsilon = 0.2$.

Task 1	Task 2	Task 3	Task 4	Task 5
0.67 ± 0.08	0.52 ± 0.10	0.56 ± 0.09	0.64 ± 0.06	0.54 ± 0.15
Task 6	Task 7	Task 8	Task 9	Task 10
0.42 ± 0.16	0.52 ± 0.08	0.65 ± 0.05	0.56 ± 0.04	0.38 ± 0.12

Table 10: The average Spearman correlation coefficients over 5 random seeds on the synthetic dataset. $\delta = 0.4$ and $\epsilon = 0$.

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
0.57 ± 0.10	0.59 ± 0.08	0.73 ± 0.05	0.76 ± 0.05	0.60 ± 0.03	0.75 ± 0.03
Task 7	Task 8	Task 9	Task 10	Task 11	Task 12
0.85 ± 0.02	0.72 ± 0.04	0.67 ± 0.06	0.50 ± 0.07	0.74 ± 0.02	0.70 ± 0.06
Task 13	Task 14	Task 15	Task 16	Task 17	Task 18
0.58 ± 0.05	0.81 ± 0.00	0.74 ± 0.03	0.69 ± 0.02	0.82 ± 0.05	0.63 ± 0.03
Task 19	Task 20	Task 21	Task 22	Task 23	Task 24
0.74 ± 0.02	0.75 ± 0.04	0.67 ± 0.04	0.63 ± 0.04	0.58 ± 0.05	0.69 ± 0.05
Task 25	Task 26	Task 27	Task 28	Task 29	Task 30
0.66 ± 0.12	0.77 ± 0.02	0.72 ± 0.06	0.60 ± 0.09	0.84 ± 0.02	0.80 ± 0.04

Table 11: The average Spearman correlation coefficients over 5 random seeds on the HAR dataset.

Leave One Out Figure 3 and Figure 4, show the results on the synthetic dataset for each task selected as the target task with different δ and ϵ. Figure 5 and Figure 6 show results when the data to be deleted are from different tasks than the tasks in the main text. The linearity relation in both cases is still preserved, meaning our MTIF align well with LOO scores.

H.1.2 The Human Activity Recognition (HAR)

The Human Activity Recognition (HAR) dataset (Anguita et al., 2013) was constructed from recordings of 30 volunteers performing various daily activities while carrying a smartphone equipped with inertial sensors on their waist. On average, each participant contributed 343.3 samples (ranging from 281 to 409). Each sample corresponds to one of six activities: walking, walking upstairs, walking downstairs, sitting, standing, or lying. The feature vectors for each sample are 561-dimensional, capturing both time and frequency domain information. We randomly select 10% of the data from each task for testing and another 10% for validation, using the remaining data to train linear models.

Leave One Task Out Table 11 shows the results on the synthetic dataset for each task selected as the target task. The correlations scores are also all very high, meaning our MTIF align well with LOTO scores in real-life datasets

H.2 Experimental Setting for Neural Networks

We train our model for 200 epochs using a StepLR learning rate scheduler with a step size of 100 and $\gamma = 0.5$. The model is optimized using cross-entropy loss and the Adam (Kingma & Ba, 2017) optimizer without weight decay, ensuring that the regularization term is zero.

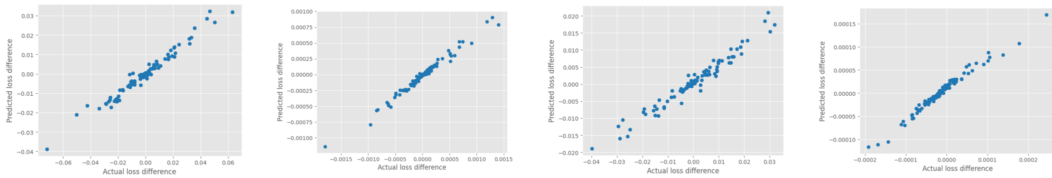

Figure 3: LOO experiments on linear regression. The x-axis is the actual loss difference obtained by LOO retraining, and the y-axis is the predicted loss difference calculated by MTIF. The first two figures from the left show within-task and between-task LOO (in order) results with $\delta = 0.4$ and $\epsilon = 0$, while the other two figures present within-task and between-task results (in order) with $\delta = 0.4$ and $\epsilon = 0.2$.

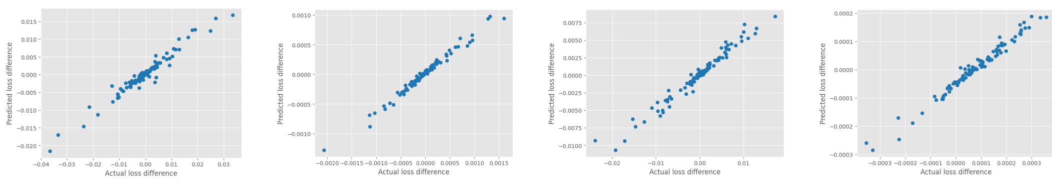

Figure 4: LOO experiments on linear regression. The x-axis is the actual loss difference obtained by LOO retraining, and the y-axis is the predicted loss difference calculated by MTIF. The first two figures from the left show within-task and between-task LOO (in order) results with $\delta = 0.8$ and $\epsilon = 0$, while the other two figures present within-task and between-task results (in order) with $\delta = 0.8$ and $\epsilon = 0.2$.

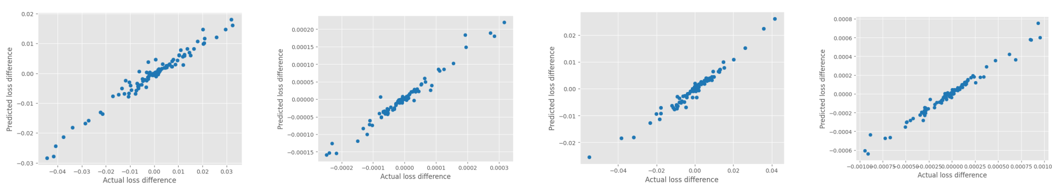

Figure 5: LOO experiments on linear regression. The x-axis is the actual loss difference obtained by LOO retraining, and the y-axis is the predicted loss difference calculated by MTIF. The first two figures from the left show within-task and between-task LOO (in order) results with deleted data from task 1, while the other two figures present within-task and between-task results (in order) with deleted data from task 2.

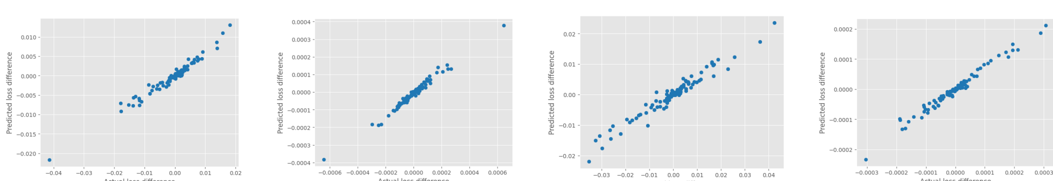

Figure 6: LOO experiments on linear regression. The x-axis is the actual loss difference obtained by LOO retraining, and the y-axis is the predicted loss difference calculated by MTIF. The first two figures from the left show within-task and between-task LOO (in order) results with deleted data from task 3, while the other two figures present within-task and between-task results (in order) with deleted data from task 5.