IDENTIFICATION OF TASK AFFINITY FOR MULTI-TASK LEARNING BASED ON DIVERGENCE OF TABULAR TASK DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task learning (MTL) can significantly improve performance by training shared models for related tasks. However, due to the risk of negative transfer between mismatched tasks, the effectiveness of MTL hinges on identifying which tasks should be learned together. In this paper, we show that for tabular datasets, this affinity between a pair of tasks can be predicted based on static features that characterize the relationship between the datasets of these tasks. Specifically, we show that we can train a regression model for predicting pairwise task affinity based on computationally-efficient features, requiring ground-truth affinity values for only a small, random sample of task pairs to generalize across all possible pairs. We demonstrate on three benchmark tabular datasets that our proposed approach can predict affinity more accurately at a lower computational cost than existing methods for identifying task affinity, which treat task data as black boxes and require training-based signals. Our work provides a practical and scalable solution to task grouping for MTL, enabling its effective application to tabular datasets with large numbers of tasks.

1 INTRODUCTION

Multi-task learning (MTL) aims to improve generalization by exploiting shared structure across similar tasks. However, its success critically depends on *which* tasks are learned together, as poorly chosen groups of tasks can lead to negative transfer. Despite the broad use of MTL across various application domains, principled methods for selecting task groups remain underexplored. The core challenge reduces to identifying *affinity* between tasks: how likely is a given pair of tasks to benefit from joint training?

Recent efforts have sought to open the black box of MTL and characterize task relationships. Data-driven methods, such as Linear Surrogate (Li et al., 2023), GRAD-TAE (Li et al., 2024), and MTGNet (Song et al., 2022), learn to predict MTL performance for groups of tasks, but they require expensive ground-truth supervision from training large numbers of MTL models. HOA (Standley et al., 2020) exhaustively measures MTL performance for all pairs of tasks, while TAG (Fifty et al., 2021) estimates affinities by repeatedly probing gradients during the training of an all-tasks MTL model. Although insightful and effective, these approaches are computationally demanding, often requiring joint training across many task groups to generate supervision labels, and they scale poorly to large sets of tasks.

Other lines of work focus on identifying the characteristics of datasets that correlate with task affinity, e.g., distributional similarity (Ben-David & Borbely, 2008) or single-task predictors of MTL affinity (Bingel & Søgaard, 2017). However, these methods stop short of directly predicting MTL performance, and they do not provide scalable mechanisms for grouping large sets of tasks.

In this paper, we propose a practical and scalable framework for predicting *pairwise MTL performance* using features that capture the *structural and statistical relationships between tabular tasks*. While MTL has seen profound success in unstructured domains (e.g., vision, NLP), *tabular data* remain the primary modality in many industry applications (e.g., finance, recommendation systems) Borisov et al. (2022). We purposefully design task-relation features that *exploit the fixed-schema nature of tabular data* (i.e., each input dimension having a consistent semantic

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

meaning across tasks), which enables us to compute lightweight, interpretable features without expensive (pre-)training or embedding extraction. Specifically, we compute pairwise task-relation features directly from dataset statistics, input-space geometry, and representation similarity; many of these can be precomputed per task and reused across pairings, which further reduces computational cost. We then train a regression model on a small, random subset of task pairs with ground-truth MTL performances—which is often orders of magnitude smaller than the subsets required by prior methods—that generalizes to all remaining pairs.

Our approach enables the scalable and inexpensive estimation of pairwise affinities even for large sets of tabular tasks, and it provides interpretable data-driven predictions that can be integrated into downstream task-grouping methods. We validate our approach on three widely used tabular MTL benchmarks, demonstrating that it accurately predicts pairwise task affinity based on a small number of training pairs. Furthermore, we show that these predictions can guide task-grouping methods to efficiently identify high-performing task groups, providing a lightweight and practical alternative to computationally-demanding prior approaches.

2 Problem Formulation

We consider a set of n tabular tasks, denoted by $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$. Each task $t \in \mathcal{T}$ is associated with a dataset $D_t = \{(x_t^{(i)}, y_t^{(i)})\}_{i=1}^{|D_t|}$, where $x_t^{(i)} \in \mathbb{R}^p$ and $y_t^{(i)}$ denote the input vector and target label, respectively. We assume a common input dimension p across all tasks. For a model with parameters θ, the prediction performance on task t is measured by a loss function:

$$L_t(D_t;\theta) \;=\; \frac{1}{|D_t|}\sum_{i=1}^{|D_t|} \ell_t\big(\hat{y}_t(x_t^{(i)};\theta),\, y_t^{(i)}\big),$$

where $\ell_t(\cdot,\cdot)$ is the task-specific positive loss (e.g., MSE or cross-entropy) and $\hat{y}_t(\cdot;\theta)$ denotes model predictions for task t. A task group is defined as a subset $G \subseteq \mathcal{T}$, which may contain between 1 and n tasks. A multi-task learning (MTL) algorithm jointly trains a model with shared parameters θ^G_{MTL} across all tasks in G by minimizing the average training loss:

$$\theta^G_{\text{MTL}} = \arg\min_{\theta} \frac{1}{|G|}\sum_{t\in G} L_t(D_t;\theta). \tag{1}$$

Similarly, θ^t_{STL} denotes the parameters of the single-task model (STL) trained independently for t:

$$\theta^t_{\text{STL}} = \arg\min_{\theta} L_t(D_t;\theta). \tag{2}$$

Task Affinity We measure *task affinity* in terms of **MTL gain**: the relative improvement in a task's loss when trained jointly with another task compared to training it independently. For a task pair $G = \{t_i, t_j\}$ with a trained MTL model $\theta^{\{t_i,t_j\}}_{\text{MTL}}$, the observed MTL gain for t_i is

$$\texttt{gain}_{t_j \to t_i} = \frac{L_{t_i}(D_{t_i};\theta^{t_i}_{\text{STL}}) - L_{t_i}(D_{t_i};\theta^{\{t_i,t_j\}}_{\text{MTL}})}{L_{t_i}(D_{t_i};\theta^{t_i}_{\text{STL}})}. \tag{3}$$

A positive value indicates that joint training with task t_j improves task t_i's performance (positive transfer), while a negative value indicates degradation (negative transfer). This definition is naturally generalized to larger task groups by replacing $\{t_i, t_j\}$ with G.

Objective: Our goal is to predict **pairwise task-affinity** (i.e., **MTL gains**) using statistical features that characterize the datasets of the tasks and their relationships, without performing exhaustive joint training over all task pairs. Formally, for any pair of tasks $\{t_i, t_j\}$ with datasets D_{t_i} and D_{t_j}, we seek to estimate gains $\texttt{gain}_{t_j \to t_i}$ and $\texttt{gain}_{t_i \to t_j}$, as defined in Equation (3), which quantify the relative improvement (or degradation) in one task's loss when trained jointly with the other. We assume access to ground-truth gains for a small subset of task pairs $\mathcal{G}_{\text{train}} \subset \binom{\mathcal{T}}{2}$, for which we perform joint MTL training. The objective is to learn a predictive model that generalizes to unseen pairs in $\mathcal{G}_{\text{test}} = \binom{\mathcal{T}}{2} \setminus \mathcal{G}_{\text{train}}$, thereby enabling scalable estimation of pairwise MTL gains.

3 AFFINITY PREDICTION WITH TASK-RELATION FEATURES

Exhaustively training MTL models for all $\binom{n}{2}$ task pairs is computationally prohibitive. So, our goal is to understand whether *pairwise MTL gain*, as defined in Equation (3), can be predicted based on statistical features that characterize the relationship between the datasets of tasks t_i and t_j. Our approach proceeds in two stages: (i) computation of task-relation features $\phi_{i,j}$ that capture structural and statistical relationships between D_{t_i} and D_{t_j}, and (ii) supervised learning of a regression function $f(\phi_{i,j};\omega)$ that maps these features to estimated MTL gains. This design ensures computational tractability: instead of performing MTL training for every possible pair of tasks (which scales quadratically with the number of tasks), we collect ground-truth gains only for a small subset of task pairs $\mathcal{G}_{\text{train}}$ and rely on a predictive model to generalize to unseen pairs.

To capture relationships between the datasets of tasks, we extract a diverse set of features quantifying statistical and structural differences, including relative dataset sizes and various distributional distances. The design of these task-relation features exploits the structured nature of tabular data, enabling us to compute them without expensive (pre-)training or embedding extraction. Each feature vector $\phi_{i,j} = \phi(D_{t_i}, D_{t_j})$ is thus computable from the task datasets directly, without performing any MTL training. The predictive function $f(\phi_{i,j};\omega)$ is trained on a limited set of ground-truth pairwise gains $\mathcal{G}_{\text{train}}$ and can efficiently estimate gains for all remaining task pairs in $\mathcal{G}_{\text{test}} = \binom{\mathcal{T}}{2} \setminus \mathcal{G}_{\text{train}}$.

While prior work has explored task affinities in transfer learning and MTL (Bingel & Søgaard, 2017; Standley et al., 2020), these studies either rely on task-specific heuristics or require repeated model evaluations across task subsets, limiting scalability. In contrast, our approach systematically considers **a broad set of task-relation features** that capture various aspects of dataset similarity and enable the **scalable prediction of pairwise MTL gains** without exhaustive training, making it applicable to larger sets of tasks.

Hypothesis We hypothesize that tasks with more similar data distributions are more likely to yield positive gains under joint training. The intuition is that similar input patterns imply shared underlying structures or decision boundaries, making shared representation learning more effective. To capture this, we leverage both statistical and structural features—such as dataset size and class imbalance—to quantify task relatedness. Our central insight is that **if two tasks appear similar in terms of their data distribution, they are more likely to benefit from being learned together.**

3.1 DESIGNING TASK-RELATION FEATURES

To enable predictive modeling of MTL gains, we construct a set of task-relation features that characterize the relationship between pairs of tasks using only dataset-level statistics of their input spaces. These features are intended to capture *intrinsic properties of each task's data distribution*, independent of any labels, trained model, or evaluation procedure. As is standard in multi-task learning, we assume that for each task the training and test sets are drawn from the same underlying data-generating distribution. Accordingly, to obtain stable estimates of task-level data characteristics, we compute input-space statistics using the union of training and test inputs. These features are computed once prior to any MTL training and remain fixed throughout all experiments. Below, we describe the categories of task-relation features used in our study.

Features based on Dataset Size Basic structural information about the task datasets provides a first-order characterization of task similarity. We consider both the **total size** of the datasets as well as the **relative size** of each dataset:

$$\texttt{Data-Size} = |D_{t_i}| + |D_{t_j}| \quad \text{and} \quad \texttt{Data-Ratio} = \frac{|D_t|}{|D_{t_i}|+|D_{t_j}|}, \quad t \in \{t_i, t_j\}. \tag{4}$$

To account for imbalance in dataset sizes, we also compute the **normalized difference in size**, which captures whether one dataset might dominate MTL training:

$$\texttt{Size-Gap} = \frac{||D_{t_i}|-|D_{t_j}||}{|D_{t_i}|+|D_{t_j}|}. \tag{5}$$

Distance- and Distribution-Based Features To quantify structural and distributional differences between tasks, we compute several features based on pairwise distances between samples. For each

task t, we define the **average intra-task pairwise distance**:

$$d_t = \mathbb{E}_{x,x' \sim D_t}\big[\|x - x'\|\big], \tag{6}$$

where $x, x' \in \mathbb{R}^p$ are independent samples drawn from D_t and $\|\cdot\|$ denotes the standard Euclidean (L2) norm in the p-dimensional input space. For tasks with large datasets, we approximate pairwise-distance expectations, d_t, using uniform random sampling of pairs, which provides an unbiased estimate and keeps computation tractable. Note that we follow the same sampling strategy for all features that are defined as expectations over sample pairs. We provide a detailed analysis of the computational costs of all task-relation features in Appendix A.2.1. Given two tasks t_i and t_j, we compute the **cross-task distance**, measuring how far samples from the two tasks lie from each other:

$$d_{(t_i+t_j)} = \mathbb{E}_{x \sim D_{t_i},\, x' \sim D_{t_j}}\big[\|x - x'\|\big]. \tag{7}$$

Normalized versions of these distance measures (e.g., $\frac{|d_{t_i} - d_{t_j}|}{d_{t_i} + d_{t_j}}$ or $\frac{d_{(t_i+t_j)}}{\sqrt{d_{t_i}\, d_{t_j}}}$) provide scale-invariant indicators of similarity in these metrics. We also compute **energy distance**, a statistical measure of dissimilarity between two input distributions, which is based on pairwise distances among samples across and within the two datasets. Energy distance between the input distributions of t_i and t_j is:

$$\texttt{Energy-Distance} = 2d_{(t_i+t_j)} - d_{t_i} - d_{t_j}, \tag{8}$$

where larger values indicate greater dissimilarity between the input distributions. To capture differences in central tendency, we also compute the **Euclidean distance between the means** of the input distributions:

$$\texttt{Feature-Mean Gap} = \|\mu_{t_i} - \mu_{t_j}\|, \tag{9}$$

where $\mu_t = \mathbb{E}_{x \sim D_t}[\mathbf{x}]$ is the mean input vector of task t. This metric reflects whether the datasets are centered around similar regions of the input space. Together, these features summarize both the spread (variance), relative positioning, and overall divergence of the two datasets in the input space.

Theoretical justification: Existing theory in domain adaptation Ben-David & Borbely (2008) establishes that the generalization error of a model trained on a source domain and applied to a target domain is bounded by the divergence between their input distributions. The `Energy-Distance` and `Feature-Mean Gap` features serve as computationally efficient proxies for this divergence. Empirically (Section 4), we find that lower values of these features correlate with positive transfer, which is consistent with the theoretical expectation that higher distributional similarity reduces the bound on generalization error and facilitates shared learning.

Representation-Based Features We further assess task similarity in terms of their raw or transformed input representations. In contrast to the distance-based features described earlier, which rely on Euclidean norms, here we replace the distance metric with an angular measure of alignment. Specifically, we compute the average **cosine similarity** between samples from two tasks:

$$\texttt{Cosine-Sim} = \mathbb{E}_{x \sim D_{t_i},\, x' \sim D_{t_j}}\left[\frac{\langle x, x' \rangle}{\|x\| \cdot \|x'\|}\right]. \tag{10}$$

Here, x and x' denote samples from D_{t_i} and D_{t_j}, respectively. **Principal component analysis (PCA)** provides another view: we compute the top-k principal components for each D_t and measure their alignment by averaging the absolute cosine similarity of the corresponding components:

$$\texttt{PCA-Align} = \tfrac{1}{k} \textstyle\sum_{\ell=1}^{k} |\langle u_\ell^{(i)}, u_\ell^{(j)} \rangle|, \tag{11}$$

where $u_\ell^{(i)}$ and $u_\ell^{(j)}$ are the ℓ-th principal components of D_{t_i} and D_{t_j}, respectively. The absolute value $|.|$ removes the sign ambiguity of PCA directions (since principal components are defined only up to a sign flip), making the measure invariant to direction reversals. We also consider a **rank-based divergence** measure, where for each input dimension $k \in \{1, \ldots, p\}$, we pool the input values of the two tasks, assign ranks to the values, and calculate the difference in mean rank between the tasks. The overall divergence is the average absolute rank gap across input dimensions:

$$\texttt{Rank-Div} = \tfrac{1}{p} \textstyle\sum_{k=1}^{p} \big|\bar{r}_{t_i}^{(k)} - \bar{r}_{t_j}^{(k)}\big|, \tag{12}$$

where $\bar{r}_t^{(k)}$ denotes the mean rank of dimension k for task t. The absolute value $|.|$ captures the magnitude of divergence, ignoring which task has higher or lower average rank. This metric reflects whether the relative ordering of samples differs systematically between tasks.

Graph-Based Features We quantify how interwoven the two datasets are in their input space by measuring their topological overlap. To measure this, we first construct a k-nearest neighbor (KNN) graph on the combined data $D_{t_i} \cup D_{t_j}$, where each sample is connected to the k nearest samples (in terms of Euclidean distance, with k chosen as a small constant such as 5 or 10). We then compute the **cross-link ratio**, the fraction of edges that connect samples from different tasks, where a higher ratio indicates stronger intermingling of the two datasets in input space:

$$\texttt{Cross-Link} = \frac{\text{Number of cross-task edges}}{\text{Total number of KNN edges}}. \tag{13}$$

Distance- and representation-based task-relation features are sensitive to the scale of the task data. To make distances comparable across dimensions and across tasks, we apply a benchmark-level normalization to the task inputs. Specifically, for each benchmark, we fit a quantile transformation (Pedregosa et al., 2011) to the concatenated inputs of all tasks, and we normalize the inputs of each task with this transformation before computing distances, cosine similarities, PCA-based alignment, and KNN graph features.

Theoretical Justification These features directly relate to the geometry of the input space and its influence on "gradient interference" during training. With hard parameter sharing, negative transfer often occurs when tasks compete for model capacity, leading to conflicting gradient updates (i.e., an update that improves one task harms another). `Cross-Link` measures topological overlap between task data; disjoint data distributions compel a shared encoder to map distinct input regions to potentially divergent representations, which can lead to conflicting gradients updates. `PCA-Align` directly assesses whether tasks share a low-dimensional subspace, which is crucial to the efficacy of shared representation learning. By measuring these properties statically, we are efficiently estimating the likelihood that the interference between tasks will be positive.

3.2 Predictive Modeling of Pairwise MTL Gains

Given access to ground-truth MTL gains for a subset of task pairs $\mathcal{G}_{\text{train}}$, we adopt a data-driven approach to predict pairwise MTL gains using the features introduced in the previous section. For each task pair, we first compute a pairwise feature vector $\phi_{i,j}$, which encodes the relationship between the datasets of the two tasks. Our goal is to learn a regression model that predicts MTL gain based on these task-pair features, providing an efficient alternative to exhaustively training for all task pairs.

Although many task-relation features are symmetric for a task pair, the target MTL gain is directional. We therefore treat each *ordered* task pair as a separate training instance and learn to predict $\texttt{gain}_{t_i \to t_j}$ and $\texttt{gain}_{t_j \to t_i}$ (i.e., two labels per unordered pair) using the same feature vector augmented with asymmetric size-ratio terms.

Polynomial Regression To capture higher-order interactions between features, we expand each feature vector using second-degree polynomial terms, denoted $\Phi(\phi_{i,j})$. We then fit a regularized regression model that minimizes the following objective over all labeled training pairs:

$$\min_{\omega} \sum_{\{t_i,t_j\} \in \mathcal{G}_{\text{train}}} \left(\texttt{gain}_{t_i \to t_j} - \omega^\top \Phi(\phi_{i,j})\right)^2 + \lambda \|\omega\|_2^2, \tag{14}$$

where $\texttt{gain}_{t_i \to t_j}$ denotes the ground-truth MTL gain for task t_j when trained with t_i, $\Phi(\phi_{i,j})$ is the quadratic feature expansion, and λ is the ℓ_2 regularization coefficient. While a linear variant provides a simpler and more interpretable baseline, the quadratic model better captures feature interactions and consistently achieves higher prediction accuracy (Appendix A.3). We therefore adopt the quadratic model as our default predictor.

4 Experiments and Results

We begin with an analysis of the proposed task-relation features on three benchmark MTL datasets, and then provide a comparison to baseline prediction approaches.

4.1 Experimental Framework

Benchmark Datasets We demonstrate our approach on three benchmark tabular datasets, which have been widely used in multi-task learning studies (Zhang & Yang, 2021). **School** (Bakker &

Heskes, 2003; Han & Zhang, 2015) involves predicting student exam scores across 139 schools ($|\mathcal{T}| = 139$), each treated as a separate regression task, using both school- and student-level features. **Chemical (MHC-I)** (Jacob et al., 2008; Zhou & Zhao, 2015) involves predicting peptide–molecule binding affinities for 35 molecules, each treated as a classification task ($|\mathcal{T}| = 35$). **Landmine** (Xue et al., 2007; Jawanpuria et al., 2015) contains data from 29 landmine fields ($|\mathcal{T}| = 29$), each treated as a task, with the goal of classifying whether a sample corresponds to a landmine. In all three benchmarks, each task has its own distinct set of data. We provide additional details and descriptive statistics on the datasets in Appendix A.1. These descriptive statistics reveal that inter-task distribution discrepancy is highest in School (label variance 0.88 ± 0.29, input distance 1.52 ± 0.24), moderate in Landmine (low label variance but higher input distance, indicating covariate shift), and lowest in Chemical (label variance 0.25, input distance 0.09 ± 0.02), which helps to contextualize the differences in task relationships and MTL performance between the benchmarks.

MTL Model Architecture and Predictor Configuration For all three benchmarks, we employ feed-forward neural networks for both STL and MTL. Note that the design of MTL architectures is not a vital component of our contribution. In fact, *our goal is orthogonal to the performance of specific MTL architectures and methods*, and our pairwise MTL gain prediction approach can be applied to and potentially benefit a wide range of MTL methods. For each benchmark, we perform a separate hyperparameter search on selected task groups to identify suitable architectures. STL models are trained first to establish task baselines, followed by MTL training on all task pairs $\binom{|\mathcal{T}|}{2}$ (9591 pairs for *School*, 595 for *Chemical*, 406 for *Landmine*) to obtain ground-truth pairwise gains. Ground-truth gains for all task pairs are computed only for the sake of experimental evaluation and correlation analysis. The proposed method itself requires labeling only a small set of sampled training pairs and can predict gains for all remaining pairs.

To train the gain predictor, we randomly sample 10–50% of task pairs as the labeled training set, and reserve the remaining $> 50\%$ for testing, ensuring evaluation on unseen task pairs. For the main experiments (Tables 1 and 2, Figures 2 and 3), we fix this proportion at 25%, which we found to provide a good balance between performance and efficiency. For Figures 6 and 7, we vary the proportion from 10–50% to show how affinity-prediction accuracy depends on the number of labeled task pairs (or equivalently, on running time, which scales with the number of labeled pairs). For the predictor, we select the ridge regularization coefficient λ via 5-fold cross-validation on $\mathcal{G}_{\text{train}}$, searching over 400 log-spaced values together with the intercept term (fit vs. omit) and solver choice (`auto`, `svd`, `cholesky`), and minimizing cross-validation MSE (Appendix A.1.3). Finally, we train higher-order task groups to assess the effectiveness of predicting gains for group selection.

4.2 Analysis of Task-Relation Features

A central question in our framework is which task-relation features are most informative for predicting pairwise MTL gains. To answer this, we first evaluate the predictive power of individual features, quantifying how well each correlates with observed MTL gains. This highlights both strong predictors and those that are redundant or noisy. Based on these results, we then select a subset of features that balances predictive power with interpretability and avoids collinearity.

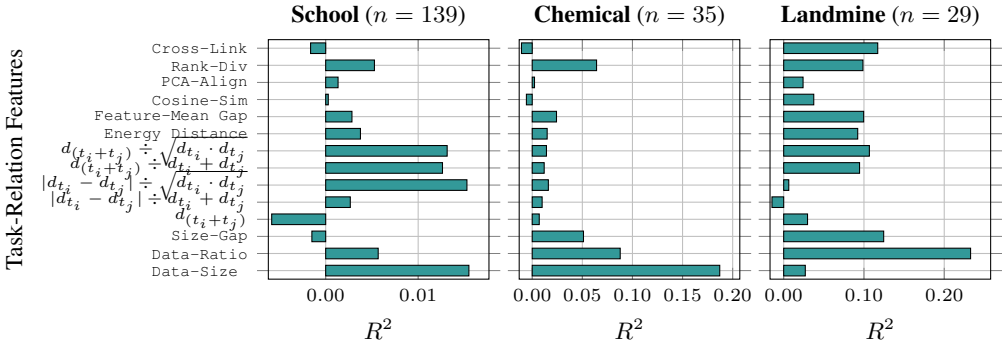


Figure 1: Predictive power of each task-relation feature in isolation, based on quadratic regression of pairwise MTL gain. Each bar represents the R^2 *value between predicted and observed MTL gain.*

Table 1: Predictive accuracy across methods for pairwise MTL gain predictions, measured as correlation between predicted pairwise affinity and actual MTL gain (higher indicates better).

Dataset	Correlation between Prediction and Actual MTL Gain		
	Inter-Task Affinity Fifty et al. (2021)	**GRAD-TAE** Li et al. (2024)	**Feature-Based** (Ours)
School	+0.002 ± 0.00	-0.002 ± 0.00	**+0.13± 0.02**
Chemical	+0.06 ± 0.04	+0.15± 0.00	**+0.50± 0.03**
Landmine	+0.34 ± 0.02	-0.26± 0.00	**+0.58± 0.03**

Predictive Power of Individual Features We train a quadratic regression model using each feature in isolation and measure R^2 between the predicted and observed pairwise MTL gains. We provide an extended analysis of the correlation between features and MTL gains in Appendix A.2.2; here, we provide results based only on predictive performance R^2 (Figure 1). Size-related features (`Data-Size`, `Data-Ratio`, `Size-Gap`) exhibit strong predictive power, particularly for *Chemical* and *Landmine*, highlighting the importance of data availability and balance between tasks. Distributional measures such as `Energy Distance` and `Feature-Mean Gap` consistently provide useful signal across all benchmarks, underscoring the crucial role of similarity between data distributions. Representation-based features (e.g., `Cosine Similarity`, `PCA Alignment`) are weak predictors on their own, though `Rank Divergence` is moderately useful in *Chemical* and *Landmine*. Finally, graph-derived feature `Cross-Link` exhibits strong predictive power in *Landmine* but inconsistent elsewhere. While predictive power varies across benchmarks, features that reflect dataset size imbalance and distributional divergence are consistently informative. This suggests that our features can serve as lightweight predictors of MTL affinity, especially when combined into a broader feature set.

Selected Subset of Features Guided by these correlation and prediction results, we curate a subset of task-relation features, which balances interpretability and predictive accuracy. The final selection includes: `Data-Size`, `Data-Ratio`, `Normalized Distances` ($|d_{t_i} - d_{t_j}| \div \sqrt{d_{t_i} \cdot d_{t_j}}$, $d_{(t_i+t_j)} \div \sqrt{d_{t_i} \cdot d_{t_j}}$), `Energy Distance`, `Feature-Mean Gap`, and `Rank Divergence`. These features span key axes of task-pair variation: *data availability* and *balance*, *distance-based similarity*, *distributional divergence*, and *structural alignment*. We excluded features that are redundant or have weak predictive power (e.g., `PCA-Align`, `Cosine-Sim`). Our selected subset of features provides a compact but informative basis for regression modeling of MTL gains. We use this subset of features in all subsequent experiments below.

4.3 Analysis of Prediction Performance

We evaluate our feature-based prediction approach in terms of both predictive accuracy and computational efficiency, comparing it to recent baselines for estimating pairwise task affinities. **TAG** (Fifty et al., 2021) estimates affinities from a single MTL model by calculating inter-task influence scores, requiring n additional forward and backward passes per update. We replicate this by training one joint model per dataset and aggregating influence scores across the training iterations. **GRAD-TAE** (Li et al., 2024) projects task gradients from a baseline MTL run and fits logistic regressions on randomly sampled task subsets. We reproduce this by sampling $2{,}000$ task subsets, estimating their performance, and defining affinity between task pair $\{t_i, t_j\}$ based on the average loss across subsets where both tasks co-occur. We evaluate both baselines in terms of correlation with ground-truth pairwise MTL gains. We provide comparisons with additional baselines in Appendix A.5.

4.3.1 Predictive Accuracy

We first compare the predictive accuracy of our feature-based approach with TAG and GRAD-TAE on the *School*, *Chemical*, and *Landmine* benchmarks. As shown in Table 1, our models consistently achieve higher correlation between predicted affinities and actual MTL gains on all benchmarks. On *School*, which is particularly challenging, our approach clearly outperforms the baselines. On *Chemical*, our approach achieves 0.50 correlation, substantially outperforming GRAD-TAE (0.15). On *Landmine*, our approach reaches 0.58, well above TAG (0.34) and GRAD-TAE (-0.26). These results highlight the efficacy of our feature-based prediction over training-intensive baselines.

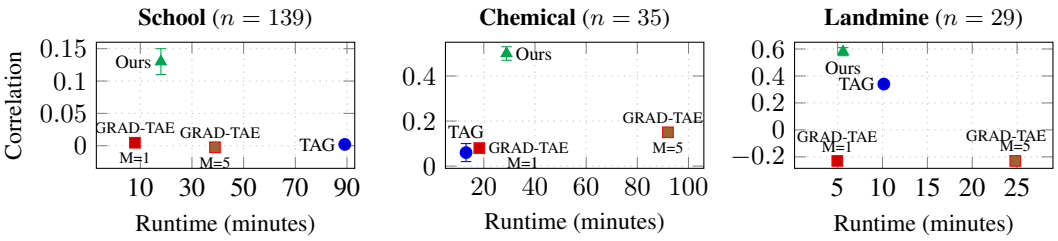


Figure 2: Correlation between the predicted and actual MTL gains (vertical axis, higher is better) and runtime (horizontal axis, lower is better) of various prediction methods. For each benchmark dataset, our method is trained with 25% of all possible pairs ($|\mathcal{G}_{\text{train}}| = 0.25\binom{n}{2}$).

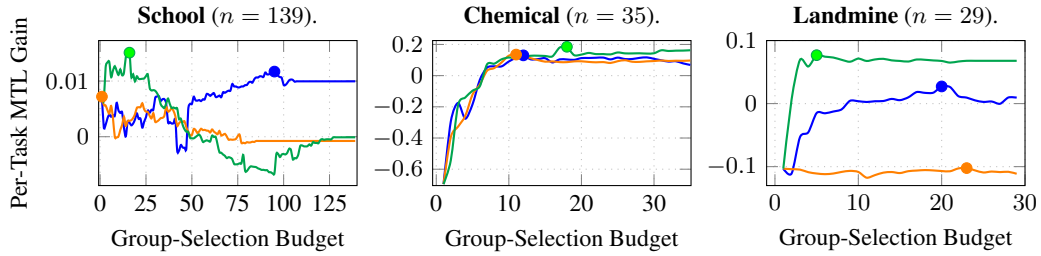


Figure 3: Comparison between prediction methods (TAG —, GRAD-TAE —, and ours —) when predictions are used for group selection with beam search, evaluated in terms of the average per-task MTL gain with the selected groups (higher maximum is better).

4.3.2 Computational Efficiency

Next, we compare runtime efficiency with the same set of baselines. The runtimes reported for our approach include MTL training for the pairs in $\mathcal{G}_{\text{train}}$ as well as feature computation, regression training, and inference. **TAG** requires one complete MTL training plus n extra forward and backward passes per gradient update to compute the $n \times n$ affinity matrix. **GRAD-TAE** requires M complete MTL trainings, $O(n)$ gradient computations and storage, and m logistic regressions on sampled subsets. This scaling incurs substantial overhead: increasing $M = 1$ to 5 nearly quintuples runtime with only marginal accuracy gains. Our approach avoids this bottleneck: after collecting ground-truth MTL gains for a small subset of task pairs $\mathcal{G}_{\text{train}}$, the remaining steps incur negligible computational cost. Crucially, while the collection of ground truths does incur substantial cost, this cost depends on the difficulty of affinity prediction and the complexity of the predictor, not on the number of tasks n (see experimental results with different training set sizes $|\mathcal{G}_{\text{train}}|$ in Figure 6 in Appendix A.4). So, our approach scales well even to large number of tasks n.

Figure 2 shows that our approach provides the best trade-off between runtime and accuracy. Using only 25% of pairs as training data, our method attains 0.13 correlation in 18 minutes on *School*, outperforming TAG (89 minutes, -0.01 correlation) and GRAD-TAE (8–39 minutes, negative correlation); on *Chemical*, it attains 0.50 in 29 minutes, significantly outperforming the baselines; on *Landmine*, it attains 0.58 in 5.6 minutes, outperforming both baselines. Overall, these results demonstrate that our framework provides a favorable balance between runtime and predictive accuracy.

4.3.3 Practical Utility in Downstream Selection of Task Groups

To evaluate the practical utility of our predicted pairwise MTL gains, we use them to select task groups. While exhaustive or branch-and-bound search could, in principle, identify globally optimal task groups (i.e., maximizing predicted gains), these methods are infeasible for our large task sets, where the number of candidate groups grows combinatorially (e.g., tens of thousands per benchmark). This motivates the use of scalable search heuristics: beam search and SDP-based clustering.

Beam Search We adopt the beam-search heuristic used by MTGNet (Song et al., 2022), which explores multiple promising groups (i.e., beams) in parallel. To estimate MTL gains for groups of three or more tasks, we aggregate pairwise MTL gain predictions in a group by averaging the predicted

Table 2: Comparison between prediction methods when predictions are used for group selection with SDP, evaluated in terms of total task loss with the selected groups (lower values are better).

		Total Loss ($\pm$ Std. Dev.) with Optimal Number of Groups k					
		TAG (Fifty et al., 2021)		**GRAD-TAE** (Li et al., 2024)		**Ours**	
Dataset	**STL Loss**	Loss	k	Loss	k	Loss	k
School	131.47	130.30 $\pm$ 0.20	3	130.74 $\pm$ 0.30	4	**130.19** $\pm$ 0.21	3
Chemical	22.98	21.64 $\pm$ 0.19	6	**20.05** $\pm$ 0.66	15	20.73 $\pm$ 0.29	2
Landmine	6.09	6.25 $\pm$ 0.15	15	6.15 $\pm$ 0.16	15	**5.70** $\pm$ 0.12	4

gains from all other tasks in a group, similar to Standley et al. (2020) and Fifty et al. (2021) (details in Appendix A.6). Once the search is finished, we perform MTL training to obtain ground-truth gains for the selected task groups. Figure 3 shows that our prediction approach consistently leads to high-performing groups in terms of MTL gains, outperforming baseline prediction approaches across all benchmarks datasets, despite the modest cost of training our pairwise MTL gain predictor.

SDP-Based Clustering We also apply semidefinite programming (SDP) for task clustering, which we adopt from Li et al. (2024). We construct a square affinity matrix from predicted pairwise MTL gains (using our approach) or from task-affinity measures (using TAG (Fifty et al., 2021) or GRAD-TAE (Li et al., 2024)). SDP-based grouping produces a desired number k of task groups. We experimented with various numbers of groups, and report the total loss of all tasks with the optimal number k in Table 2 (lower total loss is better). Across all three benchmarks, SDP-based grouping achieves competitive performance with our predicted affinities compared to TAG and GRAD-TAE, outperforming these baselines on *School* and *Landmine* and remaining competitive on *Chemical*.

5 Related Work

Understanding task affinities to maximize positive transfer and minimize interference is a core challenge in MTL. Early studies (Caruana, 1997; Argyriou et al., 2008) showed that jointly training tasks with shared structure can improve generalization. Subsequent research examined task relatedness more broadly, including zero-shot transfer (Pal & Balasubramanian, 2019), representation learning (Dwivedi & Roig, 2019), and information-theoretic perspectives (Achille et al., 2021; Zhuang et al., 2020). Surveys (Zhang & Yang, 2021; Ruder, 2017) review strategies for modeling task relationships, including shared feature representations, low-rank parameterizations, and task clustering.

The theoretical foundational work of Ben-David & Borbely (2008) formalizes task relatedness through the similarity of data-generating distributions, providing generalization guarantees but leaving practical estimation unresolved. Other studies, such as Bingel & Søgaard (2017), analyze task characteristics that correlate with transferability in specific domains but do not directly predict pairwise MTL gains. More recent methods shift focus on estimating task-affinities to grouping tasks prior to MTL training. Many rely on fully data-driven affinity estimation; for example, Standley et al. (2020) train on all task pairs to measure MTL gains, which are then aggregated to predict gains for larger groups; MTGNet (Song et al., 2022) and Linear Surrogate models (Li et al., 2023) follow a similar paradigm, requiring extensive MTL training to generalize to unseen groups. Alternatively, some approaches estimate task relatedness via a shared feature extractor for all tasks (Shiri & Sun, 2022), gradient similarity (Yu et al., 2020), or the effect of gradient updates on each task during MTL training (Fifty et al., 2021), which involves n additional forward and backward passes per gradient step. GRAD-TAE (Li et al., 2024) instead fine-tunes hyperparameters on random groups to infer performance. Though effective, these methods remain computationally expensive due to repeated MTL training or additional gradient computations and do not scale well to large sets of tasks.

6 Conclusion

We introduced a framework for predicting pairwise MTL gains for tabular datasets accurately and efficiently based on static features of task data. By quantifying divergence in task data through informative task-relation features, our method identifies task affinities with superior accuracy and minimal supervision. Experiments across multiple benchmarks show that it outperforms baselines in terms of predictive accuracy while substantially reducing runtime, demonstrating the potential of feature-driven affinity prediction for scalable and automated task-grouping for MTL.

AI Disclosure Statement

In this paper, we used generative AI tools only to check presentation issues in the manuscript, including grammar and typos in text and formulas, LaTeX formatting for tables and figures, and general language feedback. We also used these tools as a writing aid in drafting portions of the appendix, based on our own work and notes. Generative AI tools were not used to develop research ideas or methodology, design experiments, generate experimental data or results, or draw conclusions. We take responsibility for the final content of this work, including all text, claims, or artifacts produced with the aid of generative AI.

Reproducibility Statement

We provide the complete source code for our implementation and experiments as supplementary material and will release the source code under an open-source license upon publication. We provide detailed formulations and computation procedures for the task-relation features in Section 3.1 and Appendix A.2. We provide complete experimental and training configurations and hyperparameters for the underlying MTL models and the MTL gain predictor in Section 4.1 and in Appendices A.1 and A.3. Additional analyses, including computational cost, baseline comparisons, and downstream group-selection experiments, are provided in Appendices A.2, A.4, A.5, and A.6. All experiments use publicly available benchmark datasets.

Ethics Statement

This study constitutes fundamental research into machine learning, independent of any particular applications or domains. The study uses only publicly available benchmark datasets and does not involve human subjects or personally identifiable information.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016)*, pp. 265–283, 2016.

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, 2021.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73:243–272, 2008.

Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

Shai Ben-David and Reba Schuller Borbely. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73(3):273–287, 2008.

Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 164–169, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2026/.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2022.

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

Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 12387–12396, Long Beach, CA, USA, 2019. IEEE.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS 2021)*, 34:27503–27516, 2021.

Lei Han and Yu Zhang. Learning tree structure in multi-task learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 397–406, 2015.

Laurent Jacob, Jean-philippe Vert, and Francis Bach. Clustered multi-task learning: A convex formulation. *Advances in Neural Information Processing Systems (NIPS 2008)*, 21:745–752, 2008.

Pratik Kumar Jawanpuria, Maksim Lapin, Matthias Hein, and Bernt Schiele. Efficient output kernel learning for multiple tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dongyue Li, Huy L Nguyen, and Hongyang R Zhang. Identification of negative transfers in multitask learning using surrogate models. *arXiv preprint arXiv:2303.14582*, 2023.

Dongyue Li, Aneesh Sharma, and Hongyang R Zhang. Scalable multitask learning using gradient-based estimation of task affinity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1542–1553, Barcelona, Spain, 2024. Association for Computing Machinery.

Arghya Pal and Vineeth N Balasubramanian. Zero-shot task transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2189–2198, Long Beach, CA, USA, 2019. IEEE.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

Mohammad Shiri and Jiangwen Sun. Highly scalable task grouping for deep multi-task learning in prediction of epigenetic events. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1718–1723, Las Vegas, NV, USA, 2022. IEEE, IEEE.

Xiaozhuang Song, Shun Zheng, Wei Cao, James Yu, and Jiang Bian. Efficient and effective multi-task grouping via meta learning on task combinations. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:37647–37659, 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 *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pp. 9120–9132, virtual, 2020. JMLR.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8(1):35–63, 2007.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:5824–5836, 2020.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

Qiang Zhou and Qi Zhao. Flexible clustered multi-task learning by learning representative tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):266–278, 2015.

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, 2020.

A Appendix

A.1 Experimental Setup and Details

This section provides additional experimental details to facilitate reproducibility, including dataset characteristics, input preparation for the underlying MTL tasks, MTL model architectures, predictor hyperparameter selection, and the distribution of ground-truth pairwise MTL gains.

A.1.1 Dataset Preparation and Statistics

This section provides additional details on the three benchmark datasets introduced in Section 4.1, including dataset-level statistics and the preprocessing steps applied prior to MTL training. Table 3 summarizes the benchmark datasets and their characteristics.

Table 3: Benchmark tabular datasets and their characteristics.

Dataset	Task Type	Loss Metric	#Tasks	Dataset Size ($\lvert D_t \rvert$)		
				Avg.	Min.	Max.
School	Regression	Mean Squared Error (MSE)	$\lvert\mathcal{T}\rvert = 139$	110	22	251
Chemical	Classification	Binary Cross-Entropy	$\lvert\mathcal{T}\rvert = 35$	435	22	2,368
Landmine	Classification	Binary Cross-Entropy	$\lvert\mathcal{T}\rvert = 29$	511	445	690

To ensure fair comparisons in MTL performance, we equalize task dataset sizes D_t across benchmarks. Since tasks vary in training set size (Table 3), we repeat samples so that each task matches the maximum size, preventing tasks with larger training sets from contributing disproportionately to optimization. For the *Chemical* dataset, we additionally balance positive and negative samples (Jacob et al., 2008). Please note that *all task-relation features are computed on the original task datasets before any sample repetition or class balancing*. Resampling and class balancing are applied only during MTL training to stabilize optimization, so they do not affect feature construction.

Table 4: Summery of dataset statistics for pairwise MTL gain prediction study using feature based on task characteristics: label variability and within-task input distance for tasks from the three benchmark datasets.

Dataset	σ^2 (Labels)	σ (Labels)	Within-Task Distance	Observation
School	0.88 ± 0.29	0.93 ± 0.15	1.52 ± 0.24	Moderate label variability, diverse input space
Chemical	0.25	0.50	0.09 ± 0.02	**Uniform** label difficulty, tightly clustered features
Landmine	0.06 ± 0.01	0.24 ± 0.03	1.13 ± 0.09	Low label spread, input moderately diverse

Summary of Dataset Statistics: To better understand the differences in task behavior, we also investigate label variance, standard deviation, and within-task input distance for the benchmarks and summarize the key statistical properties in Table 4. In the *School* dataset, there is noticeable variability across tasks, reflected by relatively high label variance (0.88 ± 0.29) and within-task input distance (1.52 ± 0.24), indicating moderate distribution differences between tasks (e.g., between two school datasets). In contrast, the *Chemical* dataset is highly homogeneous: it has low label variance (0.25) and very small within-task distances (0.09 ± 0.02), suggesting minimal distribution discrepancy between tasks. The *Landmine* dataset exhibits low variation in label distributions (0.061 ± 0.01), but moderate diversity in input space (1.13 ± 0.009), indicating some degree of covariate shift despite similar label behavior. These observations indicate that the *School* dataset shows moderate label variability and a diverse input space. *Chemical* dataset exhibits uniform label difficulty with tightly clustered features, while *Landmine* has low label spread and moderately diverse inputs. These characteristics help contextualize differences in task behavior and model performance across datasets.

A.1.2 MTL Model Architecture

In our experiments with single-task and multi-task learning on the three benchmark datasets, we employ feed-forward neural network (NN) models. Note that the specific MTL architectures are not a vital component to our contribution. In fact, *our goal is orthogonal to the performance of specific MTL methods, and our pairwise MTL gain prediction approach can be applied to and potentially benefit a wide range of MTL methods.* We conduct a random neural architecture search on randomly chosen task groups from each of the three benchmark datasets to select the MTL architectures. We initialize the randomized architecture search with separate task-specific layers and shared layers among all the tasks. We explore variations in the number of hidden layers, number of neurons per layer, and learning rates to minimize overall loss across the tasks. Finally, we select the architecture that balances predictive accuracy and model complexity. The best architecture discovered during the search for each benchmark is adopted as the final MTL architecture. The MTL model hyperparameters for each benchmark are presented in Table 5.

Table 5: Training configuration (underlying MTL models) for benchmark datasets.

Dataset	Shared Layers	Neurons per Layer	Learning Rate α	Batch Size	Other Settings
School	3	[20, 10, 32]	0.005	64	Num. of epochs = 1000 Early stopping = 50
Chemical	2	[32, 16]	0.001	264	Num. of epochs = 1000 Early stopping = 50
Landmine	2	[64, 32]	0.001	64	Num. of epochs = 1000 Early stopping = 50

Implementation and Training: We implement neural network prediction models using Keras with TensorFlow Abadi et al. (2016), minimizing either mean squared error (*School*) or binary cross-entropy (*Chemical* and *Landmine*) with the Adam optimizer Kingma & Ba (2014).

A.1.3 Predictor Hyperparameter Search

The regularization coefficient λ in our quadratic regression predictor (Ridge penalty) controls the tradeoff between over- and underfitting. Prior to fitting, we standardize both the polynomial-expanded features $\Phi(\phi_{i,j})$ and the target gains using their training-set statistics. For each benchmark, we select λ via a grid search over 400 log-spaced values in $[10^{-10}, 10^{4}]$, combined with a search over the intercept term (fit vs. omit) and the linear-solver choice (`auto`, `svd`, `cholesky`), using 5-fold cross-validation on the labeled training pairs $\mathcal{G}_{\text{train}}$ and mean squared error as the selection criterion. The configuration with the lowest mean cross-validation MSE is selected as the final predictor for that benchmark.

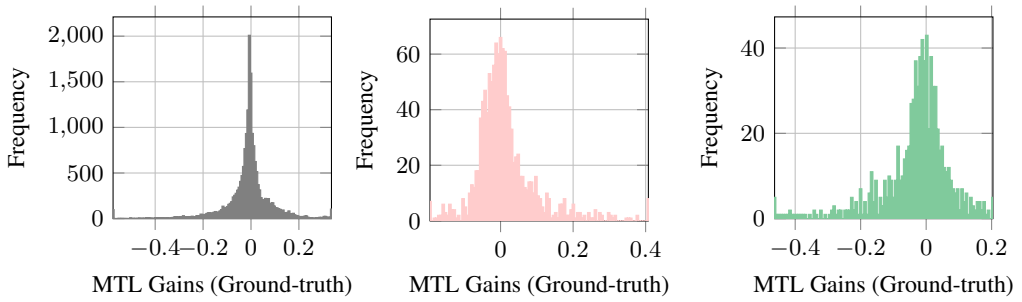
A.1.4 Distribution of Ground-Truth Pairwise MTL Gains

Before training predictive models, we visualize the empirical distribution of pairwise MTL gains for each dataset. Figure 4 shows histograms of the ground-truth MTL gains for the three benchmark datasets: *School*, *Chemical*, and *Landmine*. To remove extreme outliers and better visualize the core structure, we clip the values between the 0.5^{th} and 99.5^{th} percentiles of each dataset prior to plotting.

Across all datasets, we observe a mix of positive and negative MTL gains:

- ***School*:** Among 19,182 values (2 gains per 9,591 pairs), 44% are positive (8,422) and 56% negative (10,760), suggesting a slight dominance of negative transfer.
- ***Chemical*:** Distribution is mildly right-skewed, with 614 positive and 576 negative gains, consistent with the dataset's uniform task difficulty.
- ***Landmine*:** 344 positive vs. 468 negative gains, indicating a relatively balanced distribution with strong task incompatibilities in some pairs.

These observed gain distributions motivate the need for a robust predictive model that can distinguish between helpful and harmful task pairings before actual training.



(a) **School** ($n = 139$) (b) **Chemical** ($n = 35$) (c) **Landmine** ($n = 29$)

Figure 4: Histograms of Pairwise MTL Gains (Ground-Truth Values) for School, Chemical, and Landmine datasets

A.2 Additional Analysis of Task-Relation Features

This section provides additional analysis of the task-relation features, focusing on their computational complexity and their empirical relationship with ground-truth pairwise MTL gains.

Table 6: Theoretical computational cost of task-relation features, where $|D_{t_i}|$ and $|D_{t_j}|$ denote the number of samples in tasks t_i and t_j (Section 2), $N = |D_{t_i}| + |D_{t_j}|$, and p is the shared input dimension. Costs correspond to exact computation unless otherwise stated. For pairwise expectation-based features, sampling s pairs reduces the corresponding cost to $O(sp)$. The cost of nearest-neighbor search depends on the search algorithm and data dimensionality.

Feature	**Complexity**	**Notes**
`Data-Size`, `Data-Ratio`, `Size-Gap`	$O(1)$	Given known dataset sizes $\lvert D_{t_i}\rvert, \lvert D_{t_j}\rvert$.
Intra-task distance d_t	$O(\lvert D_t\rvert^2 p)$; or $O(sp)$ if sampled	Exact pairwise distance among a single task's $\lvert D_t\rvert$ samples, or estimated from s sampled pairs.
Cross-task distance $d_{(t_i+t_j)}$	$O(\lvert D_{t_i}\rvert \lvert D_{t_j}\rvert p)$; or $O(sp)$ if sampled	All-pairs distance between the samples of the two tasks, or estimated from s sampled cross-task pairs.
`Energy-Distance`	$O(N^2 p)$; or $O(sp)$ if sampled	Combines cross-task and both intra-task pairwise distance terms; inherits the sampled cost when each term is estimated from sampled pairs.
`Feature-Mean Gap`	$O(Np)$	Single pass to compute per-task mean vectors μ_{t_i}, μ_{t_j}.
`Cosine-Sim`	$O(\lvert D_{t_i}\rvert \lvert D_{t_j}\rvert p)$; or $O(sp)$ if sampled	All-pairs cosine similarity between samples of the two tasks, or estimated from s sampled cross-task pairs.
`PCA-Align`	$O(\lvert D_{t_i}\rvert p\, c_{t_i} + \lvert D_{t_j}\rvert p\, c_{t_j})$	Exact SVD-based PCA computed separately for each task, where $c_t = \min(n_t, p)$. If randomized/truncated PCA is used, the practical cost can be lower.
`Rank-Div`	$O(pN \log N)$	Sorting the combined samples along each of p input dimensions.
`Cross-Link` (KNN)	$O(N^2 p)$ exact; data-structure/implementation dependent approximate	Brute-force vs. tree-based / approximate nearest-neighbor search over the combined data $D_{t_i} \cup D_{t_j}$.

A.2.1 Computational Complexity of Task-Relation Features

All task-relation features described in Section 3.1 are computed directly from the input data for each task pair, without requiring any model training. Since most features are specified in closed form, their computational cost is straightforward to characterize. Table 6 summarizes the time complexity of each feature category, where $|D_{t_i}|$ and $|D_{t_j}|$ denote the number of samples in tasks t_i and t_j, $N = |D_{t_i}| + |D_{t_j}|$, and p is the shared input dimensionality.

Dataset-size features (`Data-Size`, `Data-Ratio`, `Size-Gap`) require only $O(1)$ time given known dataset sizes. Distance-based features require computing pairwise distances between samples, which costs $O(|D_{t_i}|\,|D_{t_j}|\,p)$ for the cross-task distance $d_{(t_i+t_j)}$ and $O(|D_t|^2 p)$ for the intra-task distance d_t of a task t with $|D_t|$ samples; as noted in Section 3.1, this cost can be reduced to $O(sp)$ by estimating these expectations from s randomly sampled pairs, at the expense of some estimation variance. The same sampling strategy also applies to `Energy-Distance` and `Cosine-Sim`, both of which are defined as expectations over pairs of samples (x, x') and can therefore be estimated from a random subset of s such pairs when exact computation over D_{t_i} and D_{t_j} is costly. `Feature-Mean Gap` requires only a single pass over the combined data, at $O(Np)$. Representation-based features similarly scale with the number of pairwise comparisons: `Cosine-Sim` costs $O(|D_{t_i}|\,|D_{t_j}|\,p)$ before sampling, while `PCA-Align` costs $O(|D_{t_i}|\,p\,c_{t_i} + |D_{t_j}|\,p\,c_{t_j})$, where $c_t = \min(|D_t|, p)$ for a task t, using standard SVD-based PCA computed separately per task. `Rank-Div` requires sorting the combined samples along each of the p input dimensions, incurring $O(Np \log N)$. Finally, `Cross-Link` requires constructing a k-nearest-neighbor graph over the combined data $D_{t_i} \cup D_{t_j}$; a brute-force implementation costs $O(N^2 p)$, though this can be reduced to $O(Np \log N)$ using standard approximate or tree-based nearest-neighbor search when p is not too large. Overall, the dominant cost across all feature categories is at most quadratic in the combined task-pair sample size N, and each feature can be computed independently and in parallel across task pairs.

A.2.2 Correlation Analysis Between Task-Relation Features and MTL Gains

We analyze the Pearson correlations between individual features and observed pairwise MTL gains across *School*, *Chemical*, and *Landmine* datasets, presented in Figure 5. In *School*, correlations are generally weak (all below 0.1 in magnitude), consistent with its moderate label variability and relatively homogeneous task structure—no single feature strongly explains transfer patterns. In contrast, *Chemical* and *Landmine* exhibit more distinctive trends:

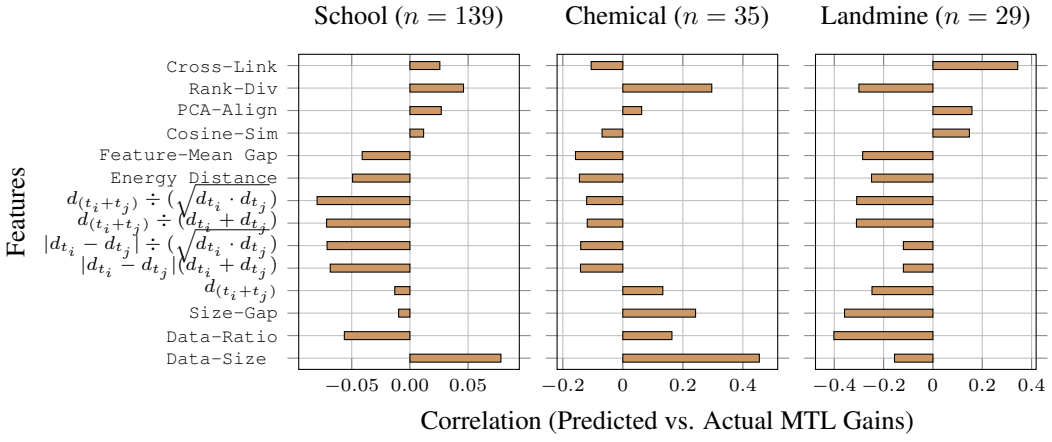


Figure 5: Correlation between task-characteristic features and actual MTL gains across three benchmark datasets.

- **Data Quantity and Balance:** In *Chemical*, size-driven features such as `Data-Size` (0.45) and `Data-Ratio` (0.16) show moderate positive correlations with gains, suggesting that task pairs with balanced and sufficiently large data are more likely to bene-

fit from MTL. In contrast, these features are weak or negative in *Landmine* (-0.40 for `Data-Ratio`), reflecting domain heterogeneity and noise sensitivity.

- **Distance and Distribution Properties:** Distance-based features exhibit consistent negative correlations in *Landmine*, with values as low as -0.31 (`normalized inter-task distance,` $d_{(t_i+t_j)} \div (d_{t_i} + d_{t_j})$), indicating that greater dissimilarity in input structure often predicts reduced transferability. `Energy Distance` and `Feature-Mean Gap` also align negatively across datasets, reinforcing the role of distributional mismatch in driving negative transfer.
- **Representation and Graph Features:** Representation-based features such as `Cosine-Sim` and `PCA-Align` have weaker, inconsistent correlations, though they are positive in *Landmine*. The graph-inspired `Cross-Link` feature shows strong positive correlation in *Landmine* (0.34) but negligible signal in *School* and *Chemical*.

Overall, these results suggest that while no single feature is universally predictive, dataset balance and distributional divergence are among the most reliable indicators—supporting our use of a learned model that can combine multiple signals.

A.3 Linear vs. Quadratic Regression for Pairwise MTL Gain Prediction

To further examine the effectiveness of our predictive framework, we conducted supplementary experiments comparing alternative modeling choices. Table 7 reports results for two variants trained on the final feature subset: a standard linear regression model and a quadratic regression model with ridge regularization. Performance is measured using R^2 scores and Pearson correlation between predicted and ground-truth MTL gains. Each model is trained on 25% of randomly selected task pairs, and results are averaged over multiple random subsets.

Table 7: Comparison of predictive accuracy across model variants for **pairwise MTL gain prediction**. Reported are mean $\pm$ std of R^2 scores and Pearson correlations between predicted and ground-truth gains.

Dataset	Feature-Based Prediction Approach			
	Linear Regression		Quadratic Regression	
	R^2	Correlation	R^2	Correlation
School	0.01$\pm$ 0.00	0.10$\pm$ 0.01	0.02$\pm$ 0.01	0.13$\pm$ 0.03
Chemical	0.20$\pm$ 0.04	0.45$\pm$ 0.05	0.22$\pm$ 0.05	0.50$\pm$ 0.03
Landmine	0.30$\pm$ 0.02	0.56$\pm$ 0.01	0.34$\pm$ 0.03	0.58$\pm$ 0.03

Across all three benchmarks, the quadratic model consistently outperforms its linear counterpart, with the largest gains observed in the *Chemical* and *Landmine* datasets. These findings validate both our feature selection and modeling strategy, demonstrating that a compact, intuitive feature set combined with a regularized quadratic regression model can effectively approximate pairwise MTL gains.

A.4 Empirical Runtime Comparison

Table 8 compares the runtime and predictive quality (correlation with observed MTL gains) of our proposed pairwise gain predictor against two strong baselines, TAG (Fifty et al., 2021) and GRAD-TAE (Li et al., 2024), across the three benchmark datasets. The results highlight two key advantages of our approach:

First, our method achieves substantially higher predictive accuracy across all datasets. For example, it reaches a correlation of $+0.52$ on *Chemical* and $+0.58$ on *Landmine*, outperforming TAG by large margins ($+0.06$ and $+0.34$, respectively) and significantly improving over GRAD-TAE, which fails to produce meaningful correlation and can even result in negative values (e.g., -0.23 on *Landmine*).

Second, our method provides a favorable trade-off between runtime and prediction quality. While TAG requires full MTL training with inter-task affinity computation throughout (e.g., 89.14 minutes on *School*), and GRAD-TAE with $M = 5$ scales linearly with the number of baseline trainings

Table 8: Correlation and Runtime between predicted pairwise affinity and actual pairwise MTL gains for different methods. Time for TAG includes a baseline MTL training with inter-task affinity estimation throughout the training. Time for Grad-TAE includes M full Baseline MTL training, plus O(n) gradient evaluations and solving logistic regression m times where m indicates the number of random subsets selected.

Dataset	TAG		GRAD-TAE				Ours ($\lvert\mathcal{G}_{train}\rvert = 0.25\binom{n}{2}$)	
			$M = 1$		$M = 5$			
	Time	Correlation	Time	Correlation.	Time	Correlation.	Time	Correlation
School	89.14	+0.002 ± 0.00	7.78	+0.005 ± 0.00	38.91	-0.002 ± 0.00	17.98	**+0.13 ± 0.02**
Chemical	13.05	+0.06 ± 0.04	18.23	+0.08 ± 0.01	91.17	+0.15 ± 0.00	28.88	**+0.52 ± 0.02**
Landmine	10.15	+0.34 ± 0.02	4.96	-0.23 ± 0.01	24.80	-0.23 ± 0.01	5.61	**+0.58 ± 0.05**

(reaching 91.17 minutes on *Chemical*), our approach only requires partial pairwise training (25% of all pairs) and converges in markedly less time (e.g., 17.98 minutes on *School* and 5.61 minutes on *Landmine*).

Overall, these demonstrate that our predictor not only delivers the strongest correlation with true pairwise gains but also does so with lower or competitive runtime, making it a more scalable and practical choice for large-scale multi-task learning settings.

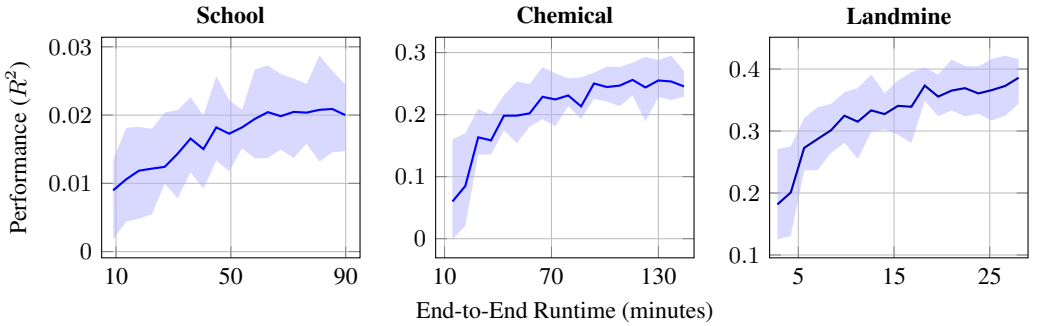


Figure 6: Prediction performance of our approach vs. runtime, with shaded bands showing the interquartile range (25th–75th percentile). Runtime includes MTL training on $\mathcal{G}_{\text{train}}$ (10–50% of all task pairs) and feature-based prediction, excluding the offline cost of labeling $\mathcal{G}_{\text{test}}$ for evaluation.

Data-Efficiency of Our Approach Figures 6 and 7 show how prediction performance improves as we increase the number of training pairs (i.e., the runtime allocated to training on the sampled pairs in $\mathcal{G}_{\text{train}}$) used to fit the gain predictor.

Across all three datasets—*School*, *Chemical*, and *Landmine*—we observe a clear upward trend: increasing the training set from 10–50% of all possible pairs steadily improves both prediction accuracy (R^2) (Figure 6) and correlation with observed MTL gains (Figure 7), while runtime grows far more slowly than for GRAD-TAE, with diminishing returns as the training set approaches 50% of all pairs.

This result demonstrates that the proposed predictor is data-efficient: even with a small fraction of available task pairs (5-10%), it achieves meaningful correlation (e.g., 0.10 for *School*, 0.45 for *Chemical*, 0.48 for *Landmine*). As training coverage grows, prediction quality steadily improves, eventually saturating around 30-40% of pairs. These findings highlight the benefit of collecting more labeled pairs for training but also show that reliable performance can be achieved without exhaustively evaluating every pair. This makes our approach practical for real-world MTL settings, where the total number of task pairs grows quadratically with the number of tasks and exhaustive evaluation is infeasible.

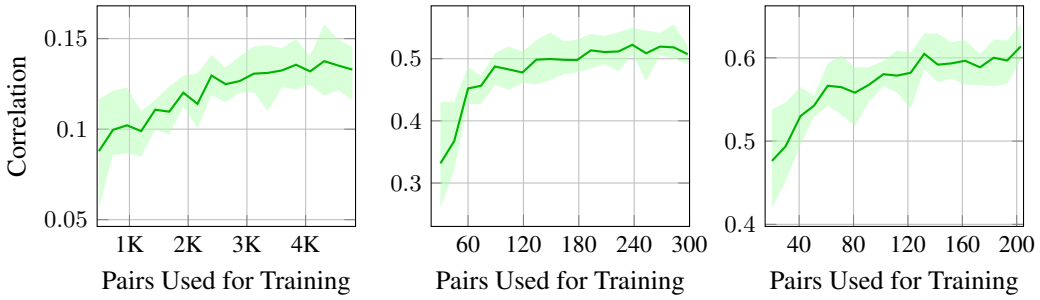


Figure 7: Prediction performance of our approach (correlation between predicted and actual MTL gains) vs. number of pairs used for training the predictor ($|\mathcal{G}_{\text{train}}|$), with shaded bands indicating the interquartile range (25th–75th percentile). Training set $\mathcal{G}_{\text{train}}$ comprises 10–50% of all task pairs.

Table 9: Performance summary of **Linear Surrogate** (Li et al., 2023) baseline: Overall metrics refer to the model evaluated on predictions for all tasks jointly, while per-task metrics report the mean and standard deviation across predictions for individual tasks.

Dataset	**Training Groups**	**Runtime (minutes)** (Prepare $\mathcal{G}_{\text{train}}$)	**Overall (for all tasks)** R^2	**Corr.**	**Avg. Per-task Metrics** **Corr ($\pm$ std)**	R^2 **($\pm$ std)**
School	50	25.6	0.165	0.42	0.003 $\pm$ 0.08	-0.86 $\pm$ 3.435
	100	51.2	0.167	0.42	0.007 $\pm$ 0.08	-0.80 $\pm$ 3.508
	200	102.4	0.152	0.40	0.014 $\pm$ 0.08	-0.83 $\pm$ 3.728
Chemical	30	161.7	-3.718	0.14	0.02 $\pm$ 0.18	-5.54 $\pm$ 7.233
	100	539.0	very neg	0.12	0.09 $\pm$ 0.20	-1.2e30 $\pm$ 2.9e30
	200	1078.0	very neg	0.04	0.16 $\pm$ 0.25	-2.1e29 $\pm$ 1.2e30
Landmine	30	18.5	-1.5	0.30	0.09 $\pm$ 0.20	-1.74 $\pm$ 1.931
	100	61.7	-9.9e26	0.05	0.17 $\pm$ 0.21	-1.8e27 $\pm$ 9.6e27
	200	123.4	-9.8e28	-0.07	0.24 $\pm$ 0.20	-6.1e29 $\pm$ 1.7e30

A.5 Additional Baseline: Linear Surrogate Model for Pairwise Gain Prediction

To compare our approach with additional baselines, we implemented a **linear surrogate model** (Li et al., 2023) that takes as input a binary task-presence vector (indicating which tasks are included in the group) and directly regresses on the observed multi-task learning (MTL) gains. The model is evaluated on only pairs, allowing for a fair comparison with our feature-based pairwise MTL gain predictor.

Table 9 summarizes the model's performance across different training set sizes. Per-task metrics report the mean and standard deviation for each task individually, whereas overall metrics aggregate predictions across all tasks. Differences between these metrics arise because tasks with very low variance or poorly predicted outcomes can disproportionately affect per-task R^2, while the overall metric smooths these effects. Similarly, per-task correlations are specific to a single task, whereas overall correlation is computed across a mixture of tasks with different scales, which can "average out" and lead to higher values.

For the *School* dataset, the linear surrogate achieves slightly higher correlation and R^2 than our feature-based predictor, although the cost in terms of runtime is much higher. However, this performance does not generalize: for *Chemical* and *Landmine*, the surrogate model drastically overfits, producing near-zero or negative correlation values and extremely large (sometimes negative) R^2 scores — clear indications of instability and non-identifiability. This behavior demonstrates that the high-dimensional 0-1 input encoding does not capture meaningful similarity information for pairwise transfer prediction, rendering the problem **non-identifiable under a purely linear mapping**.

These results emphasize the necessity of feature engineering and inductive bias in predicting MTL gains: without carefully designed inter-task features, as used in our method, the learning problem becomes ill-posed and prone to memorization rather than generalization.

A.6 Downstream Task-Group Selection Using Predicted Pairwise MTL Gains

This section provides details of the task-group selection procedures used to evaluate the downstream utility of our predicted pairwise MTL gains. We adopt existing group-selection strategies from prior work, namely beam search and SDP-based clustering, and use our predicted pairwise gains as the task-affinity input to these methods. We do not modify the underlying group-selection algorithms.

Beam Search for Task Group Selection. Beam search is a heuristic strategy that explores several promising partial solutions in parallel instead of committing to a single greedy path. The application of beam search first aggregates the pairwise predictions to estimate group-level MTL gains for sets of three or more tasks. To estimate the potential performance of larger task groups (with three or more tasks), we build upon previous work by Standley et al. (2020) and Fifty et al. (2021), which propose Higher-Order Affinity (HOA) aggregation techniques. In particular, for a given task group G and a task $t_i \in G$, we compute the group-to-task MTL gain by averaging the pairwise predicted affinities from all other tasks in the group to t_i:

$$\hat{\texttt{gain}}_{G \to t_i} = \frac{1}{|G| - 1} \sum_{t_j \in G, j \neq i} \hat{\texttt{gain}}_{t_j \to t_i} \tag{15}$$

This yields a group-level prediction vector $\hat{\boldsymbol{gain}}_G = \{\hat{gain}_{G \to t} : t \in G\}$, which we interpret as the predicted MTL gain values for the task group G. Once we have the group-level predictions, we apply the beam search algorithm on the candidate set of task-groups. At each step, the beam search keeps the top-scoring candidate groups (the beam), expands them by adding new tasks, scores the resulting groups using predicted gains, and prunes back to retain only the best candidates. This process repeats until the search budget is exhausted, after which the group with the highest estimated total MTL gain is selected. Beam size controls the tradeoff between accuracy and computational cost, allowing flexible adjustment. We adopt the grouping selection algorithm from MTGNet (Song et al., 2022), and refer readers to the original paper for a more detailed explanation.

Spectral SDP-Based Clustering for Task Grouping. To form task groups, we first construct a square affinity matrix that captures pairwise relationships between all tasks. The $n \times n$ affinity matrix can be obtained using predictions made by our approach, or from any alternative task-similarity method such as TAG (Fifty et al., 2021) or GRADTAE (Li et al., 2024).

We then solve a semidefinite programming (SDP) relaxation that seeks a matrix representation of task assignments which maximizes overall similarity while satisfying normalization and positivity constraints. The solution to this optimization problem is a continuous matrix indicating soft group memberships. Finally, we convert this soft solution into discrete task groups by applying a rounding procedure based on a similarity threshold. This produces exactly the desired number of task clusters. The group-selection procedure is adopted from prior work on SDP-based clustering (Li et al., 2024), and alternative solvers or rounding strategies could be used without changing the overall framework.