

Single-Input Multi-Output Model Merging: Leveraging Foundation Models for Dense Multi-Task Learning

Juan Garcia Giraldo
EPFL, Switzerland

juan.garciagiraldo@epfl.ch

Nikolaos Dimitriadis
EPFL, Switzerland

nikolaos.dimitriadis@epfl.ch

Ke Wang
EPFL, Switzerland

k.wang@epfl.ch

Pascal Frossard
EPFL, Switzerland
pascal.frossard@epfl.ch

Abstract

Model merging is a flexible and computationally tractable approach to merge single-task checkpoints into a multi-task model. Prior work has solely focused on constrained multi-task settings where there is a one-to-one mapping between a sample and a task, overlooking the paradigm where multiple tasks may operate on the same sample, e.g., scene understanding. In this paper, we focus on the multi-task setting with single-input-multiple-outputs (SIMO) and show that it qualitatively differs from the single-input-single-output model merging settings studied in the literature due to the existence of task-specific decoders and diverse loss objectives. We identify that existing model merging methods lead to significant performance degradation, primarily due to representation misalignment between the merged encoder and task-specific decoders. We propose two simple and efficient fixes for the SIMO setting to re-align the feature representation after merging. Compared to joint fine-tuning, our approach is computationally effective and flexible, and sheds light into identifying task relationships in an offline manner. Experiments on NYUv2, Cityscapes, and a subset of the Taskonomy dataset demonstrate: (1) task arithmetic suffices to enable multi-task capabilities; however, the representations generated by the merged encoder has to be re-aligned with the task-specific heads; (2) the proposed architecture rivals traditional multi-task learning in performance but requires fewer samples and training steps by leveraging the existence of task-specific models.

1. Introduction

Multi-task Learning (MTL), i.e., designing machine learning models capable of addressing multiple tasks

concurrently, has gained significant attention, driven by the promise of shared representations and the practical advantages of reduced memory and inference costs [1, 2]. Dense prediction tasks, such as semantic segmentation, depth estimation, and surface normal prediction, exemplify this need, particularly in applications like autonomous driving and indoor scene understanding, where efficiency and scalability are crucial [3]. However, designing multi-task systems is challenging due to the need to predefine task combinations [4, 5], limiting flexibility, and the prevalence of task conflicts, where optimizing one task hinders others [6, 7].

Foundation models [8] have emerged as a compelling solution to these MTL challenges, offering general-purpose feature representations that can be adapted to a wide range of tasks via fine-tuning. Recent advances in model merging [9, 10]—combining knowledge from diverse task-specific checkpoints into a single model—have enabled multi-task capabilities without joint training, as seen in task arithmetic [11], which leverages linear mode connectivity [12] to interpolate between independently fine-tuned models. These approaches have demonstrated notable success in structured and homogeneous settings, such as image classification with CLIP Vision Transformers (ViTs) [13, 14] or language modeling using LLaMa [15] and T5 [16].

Despite these advancements, model merging approaches for computer vision have been largely confined to simplified settings, such as solely classification tasks, which do not fully reflect the complexities of real-world multi-task applications [7, 17]. Dense prediction tasks such as semantic segmentation and depth estimation, introduce diverse objectives and heterogeneity that challenge the assumptions of existing methodologies [4]. Furthermore, existing model merging methods typically assume a constrained setting where each task operates on distinct inputs, whereas real-world applications often follow a single-input, multiple-output (SIMO) paradigm—e.g.,

predicting both semantic segmentation and depth from the same image—posing new challenges due to shared representations and interdependent task outputs.

In this paper, we extend model merging to tackle SIMO multi-task learning settings with a focus on scene understanding tasks. First, we show that existing model merging methodologies, such as Task Arithmetic [11] or TIES [9], fail in this setting and we show qualitative differences between CLIP classification [11] and our SIMO paradigm. Our approach begins with independently fine-tuning a vision foundation model paired with its corresponding lightweight head for each task, under distinct task-specific learning objectives. Subsequently, we merge the learned task-specific encoder weights into a unified shared encoder via task arithmetic, while attaching all lightweight task-specific heads. However, we observe significant performance drops due to the misalignment in the feature representations produced by the shared encoder and those of the task-specific encoders, the latter being aligned with their respective task-specific heads. To mitigate these feature distribution shifts and align the representation of the shared encoder with that of the task-specific heads, we introduce Parameter-Efficient Fine-Tuning (PEFT) strategies [18, 19], leading to substantial improvement in performance.

Our experiments are conducted on three benchmarks that exemplify diverse MTL settings: NYUv2 [20], Cityscapes [21], and a subset of Taskonomy [22]. These datasets span a range of vision tasks, including semantic segmentation, depth estimation, and surface normal prediction, among others. Using a DINOv2 backbone [23] and lightweight task-specific heads, we compare our approach to traditional MTL and state-of-the-art model merging baselines. We evaluate performance using task-specific metrics, normalized multi-task performance metrics, and visualization of task relationships, providing a comprehensive analysis of our method’s capabilities. Notably, our approach achieves competitive or superior performance to traditional multi-task learning while only requiring a lightweight adaptation after merging.

Our contributions can be summarized as follows:

- Extending model merging to single-input, multiple-output (SIMO) multi-task learning by addressing representation misalignment and task conflict through lightweight alignment between merged encoders and task-specific heads. Our method preserves task performance across diverse objectives while reducing computational overhead, as shown through comprehensive evaluations across challenging MTL settings.
- Providing insights into task relationships through task vectors, offering a novel tool for analyzing task compatibility and representation sensitivity in multi-task learning.

Overall, our approach offers a scalable and efficient alternative to joint fine-tuning, leveraging the availability of task-specific checkpoints.

2. Related Work

2.0.1 Model Merging

Directly editing models in the weight space has gained significant attention, with early works demonstrating that interpolating the weights of independently trained models often results in low-loss paths, preserving functional performance [12, 24, 25]. These findings underpin recent advancements in model merging, such as Task Arithmetic, which showed that arithmetic operations among fine-tuned weights can generate scalable multi-task capabilities [11], while theoretical insights into task arithmetic highlight weight disentanglement as a key factor for successful merging [26]. Further improvements include heuristic-guided merging strategies [27, 28], addressing parameter interference via resolving redundant updates or sign disagreements [9], preserving critical weights via the Fisher Information Matrix [29, 30], setting the merging coefficients with a linearly increasing schedule [31], and randomly dropping and rescaling the task vectors [32]. Despite the severe performance drop, the task-specific information is still encoded in the multi-task vector [10]. Ada-merging [33] and aTLAS [34] learn the merging coefficients directly from unlabeled and labeled data, respectively. However, the computer vision experiments of these works report solely on multi-task classification benchmark using an open-vocabulary model [13], while our focus lies on more challenging benchmarks of the multi-task learning literature such as scene understanding.

2.0.2 Multi-Task Learning

Learning multiple tasks withing a single model has long been a focus of machine learning research [2, 35], evolving significantly with the advent of deep learning. MTL innovations have been made by two complementary approaches; architectural modeling to combine several layers into a single cohesive backbone [36, 37] and optimization techniques [6, 7, 38, 39], focusing on the descent direction of the joint representation in a standardized shared-bottom [40] architecture. The descent direction can be computed either directly on the loss level, e.g., employing techniques such as uncertainty weighting [38], adjusting weights based on each task’s loss rate of change [41], or task relationship modeling [42], or by randomly weighting the losses [43]. Alternatively, the loss can be computed on the gradient level [39, 41], such as by resolving task gradient conflicts by aligning shared gradient directions [?], projection-based re-weighting [6], and conflict-aware gradient adjustments [17], enforcing equal projections

across all task gradients in the multi-task descent direction [7], casting the gradient combination as a bargaining game [44]. Some works have also explored weight interpolation in learning multiple tasks; to parameterize the Pareto Front [45, 46, 47] or to improve continual learning of tasks [48]. In contrast to the end-to-end joint training of the aforementioned approaches, we seek to leverage already trained single-task checkpoints to construct a multi-task model.

3. Background and SIMO Problem Statement

3.0.1 Background and Notation

In our SIMO setting we consider dense prediction tasks, such as semantical segmentation and depth estimation, for which we adopt the shared-bottom model architecture [40]. For a given task t , the task-specific model consists of an encoder with parameters θ_t and a task-specific prediction head with parameters ϕ_t . To make a prediction, the encoder produces a representation $\mathcal{Z} = f_{\text{enc}}(\mathbf{x}; \theta_t)$, which is then processed by the head to generate the prediction $\hat{\mathbf{y}}_t = g_t(\mathcal{Z}; \phi_t)$. To obtain a task-specific model for task t , we adopt a two-stage fine-tuning process following [49, 50]. In the first stage, a randomly initialized head ϕ_t is trained while keeping the encoder θ_0 frozen. In the second stage, both the encoder and the head are fine-tuned jointly.

In the multi-task setting, the model consists of a shared encoder θ_{MTL} across tasks and individual task-specific heads $\{\phi_t\}_{t=1}^T$, where the overall parameters are denoted as $(\theta_{\text{MTL}}, \phi_1, \dots, \phi_T)$ assuming a total of T tasks. During inference, the shared encoder produces a representation shared across each task $\mathcal{Z}_{\text{shared}} = f_{\text{enc}}(\mathbf{x}; \theta_{\text{MTL}})$, and the corresponding task-specific head is utilized to generate the prediction for t -th task as $\hat{\mathbf{y}}_t = g_t(\mathcal{Z}_{\text{shared}}; \phi_t)$.

3.0.2 Constructing Multi-task Model

Our goal is to construct a SIMO model capable of performing multiple tasks based on a single input. A common approach to building such a model is *multi-task learning* (MTL), where the parameters $(\theta_{\text{MTL}}, \phi_1, \dots, \phi_T)$ are trained jointly on the training set of all tasks. However, MTL is computationally expensive, and requires access to all training data.

Recently, *model merging* has emerged as an efficient and flexible alternative to construct a multi-task model. With access to task-specific checkpoints $\{(\theta_t, \phi_t)\}_{t=1}^T$, model merging combines these checkpoints to form a multi-task model with minimal training. Unlike MTL, model merging does not require access to the training data of each task; instead, it directly merges the available individual checkpoints, significantly reducing computational overhead.

Among the model merging methods, *Task Arithmetic* (TA) [11] isolates the effect of fine-tuning on each task by

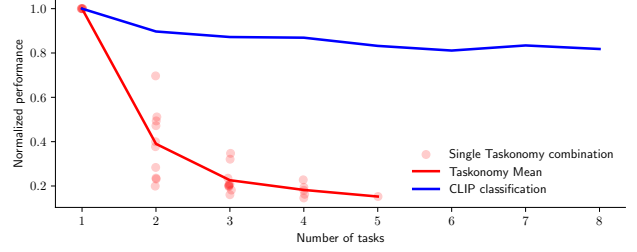


Figure 1. Comparison of performance deterioration between dense prediction (Taskonomy [22] with 5 tasks) and vision classification (8-task benchmark introduced in [11]) benchmarks as a function of the number. Each point corresponds to normalized performance of the Task Arithmetic [11] baseline for a k -task combination. Dense prediction combinations exhibit a steeper decrease compared to vision classification, indicating the increased difficulty of the setting.

operating in the space of residuals or *task vectors*, defined as $\tau_t = \theta_t - \theta_0$, $\forall t \in [T]$. These task vectors are aggregated into a multi-task vector which captures the information across tasks, $\tau_{\text{MTL}} = g(\tau_1, \tau_2, \dots, \tau_T)$, where g is an aggregation function. In task arithmetic, g is a simple summation. The final multi-task encoder weights are then obtained as $\theta_{\text{MTL}} = \theta_0 + \alpha \tau_{\text{MTL}}$, where $\alpha > 0$ is a scaling factor tuned on a held-out validation set. More recently, *Ties-Merging* (TIES) [9] follows a similar procedure to task arithmetic, while improving the merging function g to reduce the parameter interferences during merging. However, these approaches have largely been confined to simplified classification tasks with disjoint label spaces, neglecting more challenging settings like dense prediction. Such tasks, including semantic segmentation and depth estimation, involve richer supervision and spatially structured outputs that introduce greater heterogeneity and inter-task dependencies.

4. Model Merging Fails in SIMO Setting

In this section, we evaluate the effectiveness of applying model merging methods directly to construct SIMO models and show the significant performance drop of existing methods in this setting. We consider merging task-specific checkpoints on the Taskonomy dataset [22], a multi-task benchmark consisting of 5 dense-prediction tasks. Each task-specific checkpoint is fine-tuned from a DINOv2 base model [23] following the 2-stage fine-tuning process described in section 3.

After fine-tuning, we obtain the single-task checkpoints $\{(\theta_t, \phi_t)\}_{t=1}^T$, from which we construct a SIMO model with parameters $(\theta_{\text{MTL}}, \phi_1, \dots, \phi_T)$ using task arithmetic. We evaluate the normalized performance of the merged model, i.e., the percentage of performance retention w.r.t. the single-task checkpoints, while varying number of tasks being merged in Taskonomy. Figure 1 presents the normal-

ized performance¹ of the merged model against the number of task-specific checkpoints being merged, where we observe a drastic performance drop as number of tasks increases. Notably, with only 2 task-specific models being merged, the performance of task arithmetic drops by around 60% compared to the individual models, while merging the checkpoints from a total of 5 tasks results in less than 20% normalized performance. The failure of task arithmetic in this SIMO setting is in stark contrast to its success in other settings, such as merging CLIP-based image classification models [13], where it preserves almost full performance when merging 2 task-specific checkpoints [11].

5. SIMO Model Merging with Feature Re-alignment

To understand the collapsing performance of model merging methods, we start with comparing the difference in the settings primarily considered in previous model merging works and the SIMO setting. Model merging methods, such as Task Arithmetic [11] and Ties-merging [9], have primarily been applied in the context of open-vocabulary models like CLIP models [13]. CLIP’s architecture consists of a visual encoder and a text encoder that jointly learn an aligned embedding space. During fine-tuning, only the visual encoder is adapted to the target task, while the text encoder is frozen and serves as a classification head for making predictions. As a result, all individual encoders and the merged encoder remain in a consistent embedding space which is aligned with the pre-trained text encoder serving as a universal decoder for making predictions on each task. It allows the merged visual encoder to generalize across tasks without the need for explicit representation alignment for each task.

However, this assumption breaks in our SIMO setting, where each task requires an independent task-specific head that is fine-tuned alongside the encoder. In the SIMO setting, merging individual task-specific encoders produces an encoder that does not lie in the same embedding space as the ones originally used to train the task-specific decoders. This representation mismatch or *bias* [51] leads to degraded performance, as the merged encoder produces representations misaligned with the expected input distributions of the task-specific heads. Consequently, naive model merging methods fail to generalize effectively across multiple tasks in SIMO scenarios.

To alleviate the issue of representation mismatch in model merging for SIMO settings, we propose two strategies, re-aligning the head and re-aligning the joint representation. It is important to note that the single-task checkpoints have been fine-tuned on task-specific train datasets $\mathcal{D}_{\text{train}}^t, \forall t \in [T]$, while our strategies operate by assuming

access to a much smaller *multi-task* validation dataset $\mathcal{D}_{\text{val}}^{1:T}$.

5.0.1 Head Re-alignment

The merged encoder is frozen and the task-specific heads are fine-tuned separately on each task to adapt to the new representation, which allows for aligning in parallel and without the challenges of joint learning. This approach is computationally efficient; given sample $(x, y) \in \mathcal{D}_{\text{val}}^t$ of dataset t , the encoder representation $\mathcal{Z} = f_{\text{enc}}(x; \theta)$ can be pre-computed once since fine-tuning occurs solely on the task-specific heads.

5.0.2 Representation Re-alignment

Fine-tuning solely the head resolves the issue of the misalignment in the representation expected by the task-specific heads caused by merging. However, the encoder representation is still distorted, resulting in lower quality embeddings that can affect the final performance, especially when the number of tasks and hence the representation bias grows. For analysis purposes, we consider anisotropic scaling for aggregating task vectors [33, 34], where a different scaling coefficient per task and per layer is obtained via backpropagation. Differently than CLIP classification where a consistent increasing trend over the layer number is observed across tasks [31], Figure 2 shows that dense prediction tasks utilize the encoder representation differently. Therefore, merging with a global coefficient as in Task Arithmetic, distorts the representation and requires a more targeted approach.

After merging the encoders from different tasks, we employ a Parameter-Efficient Fine-Tuning [19] approach to align the encoded weights of each layer towards a joint representation. Specifically, we consider a light-weight LoRA module [18] after each repeated block in the transformer architecture [52]. Assume that each block implements a function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$. We then augment the architecture with a low-rank adapter per block, i.e., trainable matrices $\mathbf{A} \in \mathbb{R}^{r \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times r}$ for $r \ll \min\{n, m\}$. For input $x \in \mathbb{R}^n$, the modified output of the block is $f_{\text{modified}}(x) = f(x) + \mathbf{B}\mathbf{A}x$.

By introducing these alignment techniques, we enable model merging to more effectively function in multi-task SIMO settings, overcoming the representation shift that arises due to task-specific decoder training. In the following section, we demonstrate the effectiveness of the re-alignment on several benchmarks and analyze their effectiveness in improving the performance of traditional model merging methods.

¹Details on the normalized performance in Appendix B.

Table 1. Performance on NYUv2 using the DINOv2-base + linear head architecture, reported as absolute values with normalized metrics in parentheses and overall relative multi-task performance (Δ_{MTL}).

Method	Sem. Seg. [mIoU \uparrow]	Depth [aErr. \downarrow]	Surface Normal [mDist. \downarrow]	Δ_{MTL} \uparrow
Task-Specific	70.79	0.2314	18.02	-
MTL	69.01(97.5%)	0.2531(91.4%)	19.01(94.8%)	-5.4%
Task Arithmetic	65.78(92.9%)	0.4171(55.5%)	26.13(69.0%)	-27.5%
TIES	62.62(88.5%)	0.4157(55.7%)	31.38(57.4%)	-32.8%
TA + head re-align (Ours)	69.95(98.8%)	0.2524(91.7%)	19.29(93.5%)	-5.4%
TA + repr. re-align (Ours)	68.95(97.4%)	0.2539(91.1%)	18.99(94.9%)	-5.5%
TIES + head re-align (Ours)	66.25(93.6%)	0.2764(83.7%)	18.65(96.6%)	-8.7%
TIES + repr. re-align (Ours)	68.49(96.8%)	0.2576(89.8%)	18.99(94.9%)	-6.2%

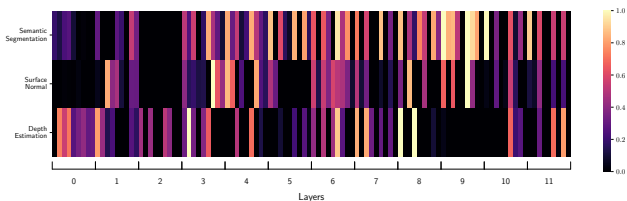


Figure 2. Optimal merging coefficients per layer as found by Adamerging [33] on NYUv2 for the DINOv2 architecture consisting of 12 repeating blocks of layers. Tasks deem important different parts of the representation; therefore applying computationally tractable but simple merging approaches like Task Arithmetic results in a representation misalignment in the encoder.

6. Experiments

In this Section, we empirically evaluate the effectiveness of our proposed approaches to adapt model merging methods to the SIMO setting. We consider two strong baselines in model merging literature, including task arithmetic [11] and Ties-merging [9], highlighting the performance improvement over them after applying our feature re-alignment methods. For the evaluation metrics, we apply different metrics for each reported task, and also report the normalized performance of the multi-task model normalized by task-specific model’s performance. As an overall assessment of the multi-task performance, we report also overall relative multi-task performance Δ_{MTL} , measuring the relative performance drop of the multi-task model. Appendix B details the evaluation metrics.

6.1. Experimental Setup

6.1.1 Benchmarks

We consider three multi-task benchmarks:

- NYUv2 [20]: 13-class semantic segmentation, monocular depth estimation, and surface normal prediction;

- Cityscapes [21]: 19-class semantic segmentation, 10-class part segmentation [53], and disparity (inverse depth) estimation
- Taskonomy [22]: Autoencoder (image compression and decompression), monocular depth estimation, surface normal prediction, SURF keypoints detection, and canny edge detection. For computational reasons, we consider a small subset of the original Taskonomy dataset, similar to [54].

The details of the evaluated benchmarks as well as the evaluation metrics are provided in Appendix A.

6.1.2 Model Architecture

We adopt DINOv2-base (86.6M parameters) as our VFM, with representation re-alignment integrating LoRA (rank 16, 0.3M parameters). For the multi-head module, we employ lightweight architectures, specifically either a linear head (≈ 0.1 M parameters) for NYUv2, or a DPT decoder (≈ 3.5 M parameters) [55] for Cityscapes and Taskonomy. Our setup follows [23] for dense recognition tasks.

6.1.3 Training Details

The single-task models required for our proposed approach, as well as the MTL models used as baselines, were trained in a two-phase process following a “probing then full fine-tuning” paradigm [49]. In the first phase, the task-specific heads were trained until convergence while keeping the backbone frozen. The second phase conducted joint fine-tuning with the backbone unfrozen until validation loss plateaued. Single-task models employed the cross-entropy loss for semantic segmentation, the dot product loss for surface normal, and the L_1 loss for all other tasks. For multi-task training, we use an unweighted mean of the losses for the individual tasks [40]. Further details regarding the hyperparameter search are provided in Appendix C.

6.1.4 Data Access

Let $\mathcal{D} = \mathcal{D}^{1:T} = \left\{ \left(\mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \dots, \mathbf{y}_T^{(i)} \right) \right\}_{i=1}^N$ be a multi-task dataset for T tasks. The multi-task baseline and single-task checkpoints have been obtained by fine-tuning on $\mathcal{D}_{\text{train}}$; single-task training has access to only one of the targets per task. Our approaches assume access to these single-task models and to a smaller multi-task validation dataset \mathcal{D}_{val} , with $|\mathcal{D}_{\text{val}}| \ll |\mathcal{D}_{\text{train}}|$.

6.2. Experimental Results

We empirically demonstrate the effectiveness of our proposed solutions for enhancing model merging techniques in SIMO settings. For each benchmark, we report the performance of the model merging methods before and after our proposed solutions, and also report the performance of single-task models and multi-task learning model as baselines.

6.2.1 Results on NYUv2

Table 1 presents the results on NYUv2. Task Arithmetic and TIES exhibit substantial performance degradation when merging task-specific models with a relative multi-task performance drop of -27.5% and -32.8% , respectively. The degradation primarily stems from misalignment between the merged encoder and the original task-specific heads, which were optimized for individual fine-tuned models. As we see from Table 1, simply applying head realignment (TA + head re-alignment, TIES + head re-alignment) significantly mitigates this issue, boosting Δ_{MTL} to normalized performance loss of only -5.4% and -8.7% , respectively. With our proposed feature re-alignment strategies to existing model merging methods, their performance is boosted to the same level as MTL (with Δ_{MTL} of -5.4%) with lower computational costs and without the need to train on a specific task combination chosen a priori.

6.2.2 Results on Cityscapes

We observe a similar trend on Cityscapes in Table 2. Direct merging via TA and TIES results in substantial performance losses with Δ_{MTL} of -31.3% and -26.1% , while re-aligning task-specific heads post-merging leads to notable improvements, with TA + head re-alignment and TIES + head re-alignment improving performance loss to only -5.3% and -5.7% , respectively. The introduction of lightweight adapters to re-align the overall representation further improves the performance of task arithmetic to -4.5% , very close to that achieved by MTL (-3.7%).

6.2.3 Results on Taskonomy

We finally evaluate the performance of our proposed methods on Taskonomy, a more challenging setting due to its diverse range of tasks. The number of tasks has increased from 3 to 5, compounding on the feature misalignment after merging as seen in Figure 1. As we observe from Table 3, Task arithmetic exhibits extreme performance degradation ($\Delta_{MTL} = -83.3\%$), while TIES also struggles at -78.0% . While the vanilla model merging methods lead to collapsing performance, our proposed feature re-alignment methods significantly mitigate the performance drop. Head re-alignment significantly improves the performance of task arithmetic and TIES to a Δ_{MTL} of -36.0% and -31.1% , respectively. Furthermore, representation re-alignment yields larger improvements, with TIES + representation re-alignment achieving the best performance at -17.4% , notably surpassing the performance of MTL with -25.7% despite much less computational costs. These findings suggest that while task diversity amplifies feature misalignment after merging, our proposed modifications such as post-hoc head fine-tuning and lightweight feature adaptation can substantially mitigate the issue. We refer to Appendix D for some qualitative results on the Taskonomy dataset.

6.3. Computational Efficiency

At training time, multi-task learning typically involves jointly optimizing multiple loss functions during training, which introduces trade-offs between tasks [4, 5, 42] and requires prior selection of task combinations. In contrast, our approach assumes the availability of pre-trained single-task checkpoints and applies a lightweight post-training alignment procedure using a small multi-task validation dataset $\mathcal{D}_{\text{val}}^{1:T}$. This method is significantly more efficient than training an MTL model from scratch on $\mathcal{D}_{\text{train}}^{1:T}$ (see Table 4), and remains efficient under changing constraints (e.g., shifts in task set, importance, or output dimensionality) as it avoids full retraining by leveraging existing single-task checkpoints.

- **Head Re-alignment:** Freezing the shared encoder allows pre-computing dataset features, training only lightweight task-specific decoders in parallel, reducing the overall computational overhead. The one-time vision encoder forward pass enables efficiency independent of encoder size, making it well-suited for scaling to billion-parameter vision models.
- **Representation Re-alignment:** Our most effective approach jointly trains a SIMO model under a multi-task objective. To mitigate the computational burden associated with standard MTL, we integrate lightweight PEFT modules into the encoder. Unlike typical MTL, only lightweight PEFT modules and task-specific

Table 2. Performance on Cityscapes using the DINOv2-base + DPT head architecture, reported as absolute values with normalized metrics in parentheses and overall relative multi-task performance (Δ_{MTL}).

Method	Sem. Seg. [mIoU \uparrow]	Part Seg. [mIoU. \uparrow]	Disparity [aErr. \downarrow]	$\Delta_{MTL} \uparrow$
Task-Specific	58.76	50.88	0.0102	-
MTL	56.36 _(95.9%)	50.36 _(99.0%)	0.0108 _(94.4%)	-3.7%
Task Arithmetic	33.56 _(57.1%)	40.55 _(79.7%)	0.0147 _(69.4%)	-31.3%
TIES	30.45 _(51.8%)	39.91 _(78.4%)	0.0111 _(91.8%)	-26.1%
TA + head re-align (Ours)	53.36 _(90.8%)	48.94 _(96.2%)	0.0105 _(97.2%)	-5.3%
TA + repr. re-align (Ours)	54.59 _(92.9%)	49.37 _(97.0%)	0.0105 _(96.7%)	-4.5%
TIES + head re-align (Ours)	52.13 _(88.7%)	48.32 _(95.0%)	0.0103 _(99.4%)	-5.7%
TIES + repr. re-align (Ours)	53.12 _(90.4%)	48.42 _(95.2%)	0.0104 _(98.1%)	-5.5%

Table 3. Performance on Taskonomy using the DINOv2-base + DPT head architecture, reported as absolute values with normalized metrics in parentheses and overall relative multi-task performance (Δ_{MTL}).

Method	Autoencoder [aErr. \downarrow]	Depth [aErr. \downarrow]	Surface Normals [aErr. \downarrow]	Edge Texture [aErr. \downarrow]	Keypoints [aErr. \downarrow]	$\Delta_{MTL} \uparrow$
Task-Specific	0.0250	0.0146	0.0696	0.0126	0.0062	-
MTL	0.0324 _(77.2%)	0.0170 _(85.9%)	0.0772 _(90.2%)	0.0213 _(59.2%)	0.0105 _(59.0%)	-25.7%
Task Arithmetic	0.2067 _(12.1%)	0.0668 _(21.9%)	0.4467 _(25.6%)	0.0750 _(16.8%)	0.0846 _(7.3%)	-83.3%
TIES	0.3132 _(8.0%)	0.3000 _(4.9%)	0.0936 _(74.4%)	0.1211 _(10.4%)	0.0513 _(12.1%)	-78.0%
TALL-Mask	0.0650 _(38.5%)	0.2271 _(6.4%)	0.0768 _(90.6%)	0.0782 _(16.1%)	0.0196 _(31.6%)	-63.4%
TA + head re-align (Ours)	0.0301 _(83.1%)	0.0270 _(54.1%)	0.1428 _(48.7%)	0.0256 _(49.2%)	0.0073 _(84.9%)	-36.0%
TA + repr. re-align (Ours)	0.0283 _(88.3%)	0.0186 _(78.5%)	0.0968 _(71.9%)	0.0199 _(68.3%)	0.0071 _(87.3%)	-21.1%
TIES + head re-align (Ours)	0.0266 _(94.0%)	0.0290 _(50.3%)	0.1707 _(40.8%)	0.0213 _(59.2%)	0.0060 _(100.3%)	-31.1%
TIES + repr. re-align (Ours)	0.0283 _(88.3%)	0.0167 _(87.4%)	0.0903 _(77.1%)	0.0195 _(64.6%)	0.0065 _(95.4%)	-17.4%

Table 4. Efficiency breakdown by method on the Taskonomy dataset. The reported metrics for our feature realignment techniques (head re-align and repr. re-align) rely on Task Arithmetic as the task merging strategy.

Method	Steps	Params	$ \mathcal{D}^{1:T} $
Task-Specific	21.8k	90.1M	16k
MTL	90.0k	103.9M	16k
TA + head re-align	14.9k	17.3M	4k
TA + repr. re-align	12.1k	17.6M	4k

heads are trained, minimally increasing parameter count and reducing computational demands.

While optimizing single-task models is generally more tractable than end-to-end MTL, the scalability of our method is limited if such single-task checkpoints are not already available, as training them individually for a large number of tasks can be costly.

6.4. Identifying Task Relationships

As a byproduct of leveraging task vectors to construct our SIMO models, we propose a simple yet novel method for identifying task relationships, e.g., for observing which tasks can be mutually beneficial or interfering. Given a primary task t , we aim to understand how adding its task-specific information to a pre-trained model affects other tasks. To achieve this, we initialize a SIMO model comprising of a pre-trained vision encoder θ_0 and fine-tuned task-specific heads (ϕ_1, \dots, ϕ_T) , assuming a total of T tasks. We then perturb the shared encoder by adding information from a single primary task $\lambda \cdot \tau_t$, where λ is chosen (e.g., $\lambda \in [0, 1.5]$). The core idea is to then evaluate how task-specific metrics vary with λ , perturbing the encoder’s feature representation toward task t , thereby exposing alignment or conflicts among the different $T - 1$ tasks.

Figure 3 illustrates our approach by analyzing task relationships within the Cityscapes dataset, providing two key insights:

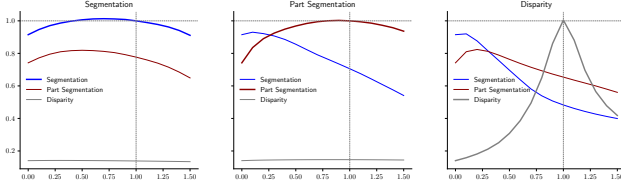


Figure 3. Visualization of task relationships for Cityscapes. Each plot presents the normalized performance on each dataset as a function of the task vector added to the model with a coefficient of the x-axis, forming the model $\theta_0 + \lambda\tau$, for $\tau \in \{\tau_{seg}, \tau_{part_seg}, \tau_{disp}\}$.

1. *task relationships*: we assess how the performance of secondary tasks evolves as the primary task is introduced. Beneficial interactions emerge when the performance of the secondary task improves, e.g., semantic segmentation enhances part segmentation. In contrast, conflicting interactions results in consistent degradation of secondary task performance, e.g., part segmentation undermining semantic segmentation;
2. *sensitivity of the feature representations*: certain tasks, such as disparity, show high sensitivity to perturbations in the encoder weights, requiring $\lambda \approx 1$ to recover optimal performance. Therefore, naively merging such sensitive weights may result in information loss.

These observations provide critical insights into task compatibility and their potential for effective multi-task learning. Further visualizations of task relationships for the NYUv2 and Taskonomy dataset are provided in Appendix E.

7. Conclusion

This study demonstrates that single-input multiple-output models can be efficiently built by leveraging single-task checkpoints and properly applying model merging techniques. By mitigating representation bias in the encoder and aligning task-specific heads to the new feature representation, our approach achieves competitive performance at a lower computational cost compared to standard multi-task fine-tuning. Experiments on large scale benchmarks, such as NYUv2, Cityscapes and Taskonomy confirm the effectiveness of the proposed method. These results suggest a promising alternative to expensive joint training, enabling scalable and flexible multi-task systems that can be assembled post hoc from existing models.

References

- [1] S. Ruder, “An overview of multi-task learning in deep neural networks,” *CoRR*, vol. abs/1706.05098, 2017. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.eprinttype: arXiv](https://arxiv.org) [tex.timestamp: Mon, 13 Aug 2018 01:00:00 +0200](https://dblp.org). 1
- [2] M. Crawshaw, “Multi-task learning with deep neural networks: A survey,” *CoRR*, vol. abs/2009.09796, 2020. [arXiv: 2009.09796](https://arxiv.org) [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 23 Sep 2020 15:51:46 +0200](https://dblp.org). 1, 2
- [3] S. Vandenhende, S. Georgoulis, W. V. Gansbeke, M. Proesmans, D. Dai, and L. V. Gool, “Multi-task learning for dense prediction tasks: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3614–3633, 2022. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Tue, 28 Jun 2022 01:00:00 +0200](https://dblp.org). 1
- [4] T. Standley, A. Zamir, D. Chen, L. J. Guibas, J. Malik, and S. Savarese, “Which tasks should be learned together in multi-task learning?,” in *International conference on machine learning (ICML)*, Proceedings of machine learning research, 2020. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Fri, 12 Aug 2022 01:00:00 +0200](https://dblp.org). 1, 6
- [5] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn, “Efficiently Identifying Task Groupings for Multi-Task Learning,” in *Advances in Neural Information Processing Systems*, vol. 34, pp. 27503–27516, Curran Associates, Inc., 2021. 1, 6
- [6] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Gradient Surgery for Multi-Task Learning,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, Curran Associates, Inc., 2020. 1, 2
- [7] L. Liu, Y. Li, Z. Kuang, J.-H. Xue, Y. Chen, W. Yang, Q. Liao, and W. Zhang, “Towards Impartial Multi-task Learning,” Oct. 2020. 1, 2, 3
- [8] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. v. Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kudipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Nieves, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, “On the Opportunities and Risks of Foundation Models,” July 2022. [arXiv:2108.07258 \[cs\]](https://arxiv.org). 1
- [9] P. Yadav, D. Tam, L. Choshen, C. A. Raffel, and M. Bansal, “TIES-merging: Resolving interference when merging models,” in *Advances in neural information processing systems (NeurIPS)*, 2023. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Fri, 01 Mar 2024 00:00:00 +0100](https://dblp.org). 1, 2, 3, 4, 5, 13
- [10] K. Wang, N. Dimitriadis, G. Ortiz-Jiménez, F. Fleuret, and P. Frossard, “Localizing task information for improved model merging and compression,” in *International conference on machine learning (ICML)*, 2024. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Mon, 02 Sep 2024 16:45:29 +0200](https://dblp.org). 1, 2, 13
- [11] G. Ilharco, M. T. Ribeiro, M. Wortsman, L. Schmidt, H. Hajishirzi, and A. Farhadi, “Editing models with task arithmetic,” in *International conference on learning representations (ICLR)*, 2023. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 24 Jul 2024 01:00:00 +0200](https://dblp.org). 1, 2, 3, 4, 5
- [12] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, “Linear mode connectivity and the lottery ticket hypothesis,” in *International conference on machine learning (ICML)*, Proceedings of machine learning research, 2020. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 10 Feb 2021 00:00:00 +0100](https://dblp.org). 1, 2
- [13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin,

- J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning (ICML)*, Proceedings of machine learning research, 2021. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 25 Aug 2021 01:00:00 +0200](https://dblp.org). 1, 2, 4
- [14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International conference on learning representations (ICLR)*, 2021. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 23 Jun 2021 17:36:39 +0200](https://dblp.org). 1
- [15] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “LLaMA: Open and Efficient Foundation Language Models,” Feb. 2023. 1
- [16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, pp. 140:1–140:67, 2020. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 11 Sep 2024 01:00:00 +0200](https://dblp.org). 1
- [17] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu, “Conflict-Averse Gradient Descent for Multi-task learning,” in *Advances in Neural Information Processing Systems*, vol. 34, pp. 18878–18890, Curran Associates, Inc., 2021. 1, 2
- [18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International conference on learning representations (ICLR)*, 2022. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Sat, 20 Aug 2022 01:00:00 +0200](https://dblp.org). 2, 4
- [19] N. Houlsby, A. Giurghi, S. Jastrzebski, B. Morrone, Q. d. Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-Efficient Transfer Learning for NLP,” June 2019. [arXiv:1902.00751 \[cs\]](https://arxiv.org/abs/1902.00751). 2, 4
- [20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor Segmentation and Support Inference from RGBD Images,” in *Computer Vision – ECCV 2012* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), vol. 7576, pp. 746–760, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. Series Title: Lecture Notes in Computer Science. 2, 5
- [21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Sat, 30 Sep 2023 01:00:00 +0200](https://dblp.org). 2, 5
- [22] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Fri, 24 Mar 2023 00:00:00 +0100](https://dblp.org). 2, 3, 5
- [23] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, “DINOv2: Learning Robust Visual Features without Supervision,” *Transactions on Machine Learning Research*, July 2023. 2, 3, 5
- [24] T. Garipov, P. Izmailov, D. Podoprikin, D. P. Vetrov, and A. G. Wilson, “Loss surfaces, mode connectivity, and fast ensembling of dnn,” in *Advances in neural information processing systems (NeurIPS)*, 2018. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Mon, 16 May 2022 15:41:51 +0200](https://dblp.org). 2
- [25] F. Draxler, K. Veschgini, M. Salmhofer, and F. A. Hamprecht, “Essentially no barriers in neural network energy landscape,” in *International conference on machine learning (ICML)*, Proceedings of machine learning research, 2018. [tex.bibsource: dblp computer science bibliography, https://dblp.org](https://dblp.org) [tex.timestamp: Wed, 29 Apr 2020 01:00:00 +0200](https://dblp.org). 2
- [26] G. Ortiz-Jiménez, A. Favero, and P. Frossard, “Task arithmetic in the tangent space: Improved editing of pre-trained models,” in *Advances in neural information processing systems (NeurIPS)*, 2023.

- tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 01 Mar 2024 00:00:00 +0100. 2
- [27] M. Davari and E. Belilovsky, “Model breadcrumbs: Scaling multi-task model merging with sparse masks,” in *IEEE european conference on computer vision (ECCV)*, Lecture notes in computer science, 2024. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 08 Nov 2024 20:51:06 +0100. 2
- [28] X. Jin, X. Ren, D. Preotiuc-Pietro, and P. Cheng, “Dataless knowledge fusion by merging weights of language models,” in *International conference on learning representations (ICLR)*, 2023. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Wed, 24 Jul 2024 01:00:00 +0200. 2
- [29] M. Matena and C. Raffel, “Merging models with fisher-weighted averaging,” in *Advances in neural information processing systems (NeurIPS)*, 2022. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Mon, 08 Jan 2024 00:00:00 +0100. 2
- [30] D. Tam, M. Bansal, and C. Raffel, “Merging by matching models in task subspaces,” *CoRR*, vol. abs/2312.04339, 2023. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.eprinttype: arXiv tex.timestamp: Tue, 02 Jan 2024 00:00:00 +0100. 2
- [31] K. Wang, N. Dimitriadis, A. Favero, G. Ortiz-Jimenez, F. Fleuret, and P. Frossard, “LiNeS: Post-training Layer Scaling Prevents Forgetting and Enhances Model Merging,” Oct. 2024. 2, 4
- [32] L. Yu, B. Yu, H. Yu, F. Huang, and Y. Li, “Language models are super mario: Absorbing abilities from homologous models as a free lunch,” in *International conference on machine learning (ICML)*, 2024. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 14 Feb 2025 00:00:00 +0100. 2, 13
- [33] E. Yang, Z. Wang, L. Shen, S. Liu, G. Guo, X. Wang, and D. Tao, “AdaMerging: Adaptive model merging for multi-task learning,” in *International conference on learning representations (ICLR)*, 2024. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Mon, 10 Feb 2025 00:00:00 +0100. 2, 4, 5
- [34] F. Z. Zhang, P. Albert, C. Rodriguez-Opazo, A. v. d. Hengel, and E. Abbasnejad, “Knowledge Composition using Task Vectors with Learned Anisotropic Scaling,” Nov. 2024. 2, 4
- [35] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.eprinttype: arXiv tex.timestamp: Mon, 13 Aug 2018 01:00:00 +0200. 2
- [36] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, “Cross-stitch networks for multi-task learning,” in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 24 Mar 2023 00:00:00 +0100. 2
- [37] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, “Latent multi-task architecture learning,” in *AAAI conference on artificial intelligence (AAAI)*, 2019. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Mon, 04 Sep 2023 12:29:24 +0200. 2
- [38] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 24 Mar 2023 00:00:00 +0100. 2
- [39] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, “GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks,” in *International conference on machine learning (ICML)*, Proceedings of machine learning research, 2018. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 13 Sep 2019 01:00:00 +0200. 2
- [40] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 10 Mar 2023 00:00:00 +0100. 2, 3, 5
- [41] S. Liu, E. Johns, and A. J. Davison, “End-To-End Multi-Task Learning With Attention,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), pp. 1871–1880, IEEE, June 2019. 2
- [42] S. Liu, S. James, A. J. Davison, and E. Johns, “Autolambda: Disentangling dynamic task relationships,” vol. 2022, 2022. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 19 May 2023 01:00:00 +0200. 2, 6

- [43] B. Lin, F. Ye, Y. Zhang, and I. W. Tsang, “Reasonable Effectiveness of Random Weighting: A Litmus Test for Multi-Task Learning,” July 2022. arXiv:2111.10603 [cs]. 2
- [44] A. Navon, A. Shamsian, I. Achituve, H. Maron, K. Kawaguchi, G. Chechik, and E. Fetaya, “Multi-Task Learning as a Bargaining Game,” in *Proceedings of the 39th International Conference on Machine Learning*, pp. 16428–16446, PMLR, June 2022. ISSN: 2640-3498. 3
- [45] N. Dimitriadis, P. Frossard, and F. Fleuret, “Pareto Manifold Learning: Tackling multiple tasks via ensembles of single-task models,” in *International conference on machine learning (ICML)*, Proceedings of machine learning research, 2023. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Mon, 28 Aug 2023 17:23:08 +0200. 3
- [46] N. Dimitriadis, P. Frossard, and F. Fleuret, “Pareto low-rank adapters: Efficient multi-task learning with preferences,” in *The thirteenth international conference on learning representations*, 2025. 3
- [47] A. Ramé, G. Couairon, C. Dancette, J.-B. Gaya, M. Shukor, L. Soulier, and M. Cord, “Rewarded soups: Towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards,” in *Advances in neural information processing systems (NeurIPS)*, 2023. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Fri, 01 Mar 2024 00:00:00 +0100. 3
- [48] S.-I. Mirzadeh, M. Farajtabar, D. Görür, R. Pascanu, and H. Ghasemzadeh, “Linear mode connectivity in multitask and continual learning,” in *International conference on learning representations (ICLR)*, 2021. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Wed, 23 Jun 2021 17:36:39 +0200. 3
- [49] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, “Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution,” Feb. 2022. arXiv:2202.10054 [cs]. 3, 5
- [50] A. Hazimeh, A. Favero, and P. Frossard, “Task Addition and Weight Disentanglement in Closed-Vocabulary Models,” 2024. 3
- [51] E. Yang, L. Shen, Z. Wang, G. Guo, X. Chen, X. Wang, and D. Tao, “Representation Surgery for Multi-Task Model Merging,” in *Proceedings of the 41st International Conference on Machine Learning*, pp. 56332–56356, PMLR, July 2024. ISSN: 2640-3498. 4
- [52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems (NeurIPS)*, 2017. tex.bibsource: dblp computer science bibliography, <https://dblp.org> tex.timestamp: Thu, 21 Jan 2021 13:58:27 +0100. 4
- [53] D. de Geus, P. Meletis, C. Lu, X. Wen, and G. Dubbelman, “Part-aware panoptic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5485–5494, 2021. 5
- [54] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, “MultiMAE: Multi-modal multi-task masked autoencoders,” in *Computer vision – ECCV 2022: 17th european conference, tel aviv, israel, october 23–27, 2022, proceedings, part XXXVII*, (Tel Aviv, Israel and Berlin, Heidelberg), pp. 348–367, Springer-Verlag, 2022. Number of pages: 20. 5
- [55] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision Transformers for Dense Prediction,” pp. 12179–12188, 2021. 5
- [56] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, “Attentive single-tasking of multiple tasks,” in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2019. 13

A. Details on evaluation benchmarks

- NYUv2: the training set comprises 795 samples with an image size of 280×378 , while the validation set contains 400 samples and the test set includes 254 samples, both maintaining the same image dimensions.
- Cityscapes: the training set consists of 2,380 samples with an image size of 128×256 , the validation set includes 595 samples, and the test set comprises 500 samples, all with identical dimensions.
- Our small subset of the Taskonomy dataset: the training set contains 16,000 samples, drawn from eight buildings (allensville, coffeen, collierville, leonardo, merom, pinesdale, ranchester, stockman) with a maximum of 2,000 samples per building and an image size of 256×256 . The validation and test sets each include 4,000 samples, sourced from four buildings each (beechwood, corozal, klikitat, shelbyville for validation; ihlen, mcdade, onaga, tolstoy for test), with a maximum of 1,000 samples per building, maintaining the same image dimensions.

B. Evaluation Metrics

B.0.1 Task-specific metrics

We assess the performance of our dense prediction tasks across all datasets using three primary metrics: (1) mean Intersection over Union (mIoU), which quantifies the overlap between predicted and ground-truth regions, (2) mean angular distance (mDist), which measures the average angular deviation (in degrees) between predicted and ground-truth values, and (3) absolute error (aErr), defined as the L1 difference between predictions and ground truth. For NYUv2 and Cityscapes we evaluate segmentation tasks via mIoU, surface normals via mDist., and depth/disparity estimation via aErr. For the Taskonomy dataset, all tasks are evaluated using the aErr.

B.0.2 Normalized performance

Assuming limited data availability and computational budget, and similar to prior model merging works [9, 32], we measure the drop in performance relative to the single-task case as a proxy for retention. Formally, let M denote a given metric for task t from the model m being evaluated, and b represents the single-task baselines.

$$\tilde{M} = \frac{1}{T} \sum_{t=1}^T \left(\frac{M_{m,t}}{M_{b,t}} \right)^{(-1)^{l_t}} \quad (1)$$

where l_t is set to 1 if lower values of $M_{m,t}$ indicate better performance, and 0 otherwise.

B.0.3 Relative multi-task performance

Following [10, 56], we also report the overall relative multi-task performance Δ_{MTL} , which measures the relative performance drop of the multi-task model compared to the individual single-task models:

$$\Delta_{MTL} = \frac{1}{T} \sum_{t=1}^T (-1)^{l_t} \frac{M_{m,t} - M_{b,t}}{M_{b,t}} \quad (2)$$

Overall, our objective is to minimize the metrics \tilde{M} and Δ_{MTL} .

C. Training Details

We detail the training hyperparameters used across all experiments, employing DINOv2-B as the encoder with the following output layer $\{3, 6, 9, 12\} + \text{CLS}$ token used by the task-specific heads (linear projections or DPT architecture) for the NYUv2, Cityscapes, and Taskonomy datasets. We adopt the AdamW optimizer with a batch size of 16 for NYUv2 and Cityscapes, and 32 for Taskonomy. The learning rate hyperparameter tuning is conducted via a Bayesian optimization technique with a log-uniform distribution ranging from 1×10^{-8} to 1×10^{-3} . We employ a once-cycle learning rate scheduler, with a warm-up ratio of 0.05 and gradient clipping set to 25.0. Our training includes a maximum of 50 epochs for task-specific models, 200 epochs for multi-task learning, and 200 epochs for our proposed representation re-alignment approaches.

D. Qualitative Results: Taskonomy

Figure 4 visualizes the results of MTL (serving as our baseline), standard model merging techniques (e.g. Task Arithmetic), and our proposed representation re-alignment approach, with our figure clearly illustrating the shortcomings of standard model merging techniques. Although Table 3 quantitatively suggests that MTL and our approach may appear suboptimal based on normalized and relative multi-task performance metrics, these visualizations highlight that one should not consider them as ground truths. Highlighting the limitations of using the absolute error (aErr) commonly used for the Taskonomy dataset to report multi-task dynamics.

E. Visualizing Task Relationships

We follow to the methodology outlined in Section 6.4 and present visualizations of task relationships for the NYUv2 and Taskonomy datasets. Figure 5 exhibits a consistent pattern with the findings in Section 6.4, whereas Figure 6 reveals that Taskonomy tasks lack overlapping feature spaces and are highly sensitive to perturbations in feature representations. This explains the poor performance of

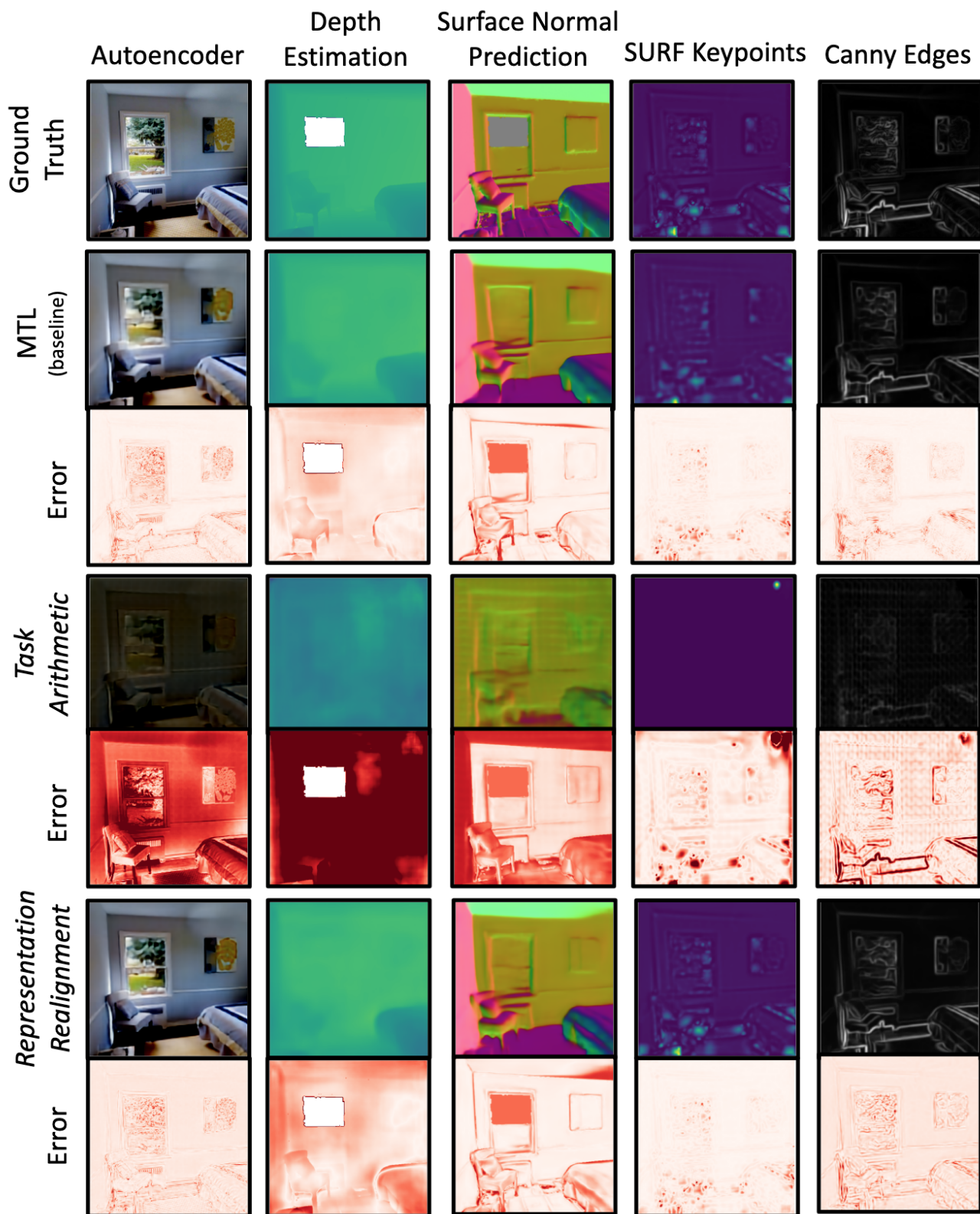


Figure 4. Qualitative results comparing MTL, Task Arithmetic, and our Representation Re-alignment technique on the Taskonomy dataset.

naive model merging techniques, which exhibit a degradation of approximately 80%.

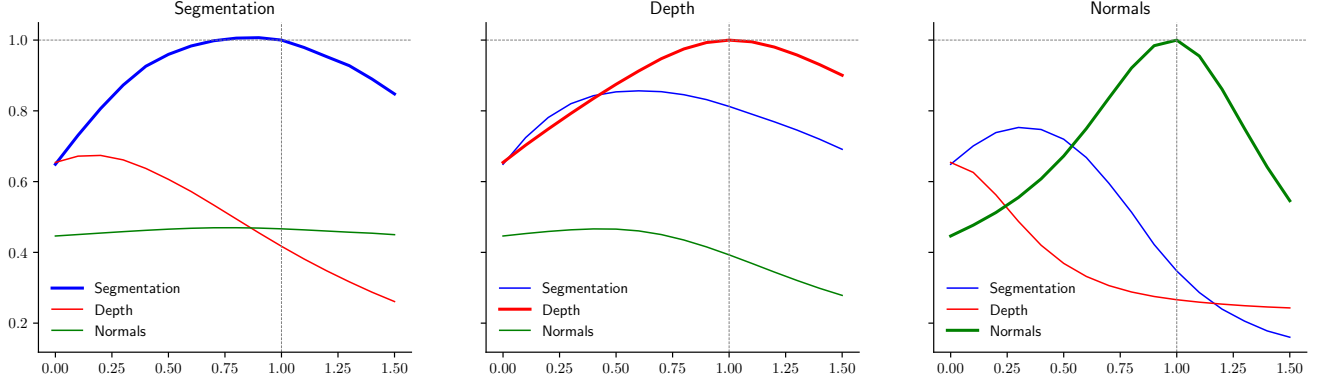


Figure 5. Visualization of task relationships for NYUv2. Each plot presents the normalized performance on each dataset as a function of the task vector added to the model with a coefficient of the x-axis, forming the model $\theta_0 + \lambda\tau$, for $\tau \in \{\tau_{seg}, \tau_{depth}, \tau_{normals}\}$.

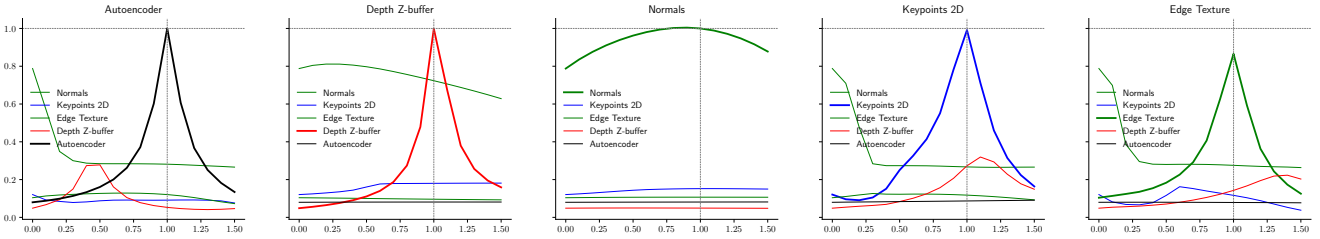


Figure 6. Visualization of task relationships for Taskonomy. Each plot presents the normalized performance on each dataset as a function of the task vector added to the model with a coefficient of the x-axis, forming the model $\theta_0 + \lambda\tau$, for $\tau \in \{\tau_{auto}, \tau_{depth}, \tau_{normals}, \tau_{keypoints}, \tau_{edges}\}$.