Rep-MTL: Unleashing the Power of Representation-level Task Saliency for Multi-Task Learning

Zedong Wang[1] Siyuan Li[2] Dan Xu[1,✉]

[1]The Hong Kong University of Science and Technology [2]Zhejiang University

zedong.wang@connect.ust.hk, lisiyuan@westlake.edu.cn, danxu@cse.ust.hk

Abstract

Despite the promise of Multi-Task Learning in leveraging complementary knowledge across tasks, existing multi-task optimization (MTO) techniques remain fixated on resolving conflicts via optimizer-centric loss scaling and gradient manipulation strategies, yet fail to deliver consistent gains. In this paper, we argue that the shared representation space, where task interactions naturally occur, offers rich information and potential for operations complementary to existing optimizers, especially for facilitating the inter-task complementarity, which is rarely explored in MTO. This intuition leads to Rep-MTL, which exploits the representation-level task saliency to quantify interactions between task-specific optimization and shared representation learning. By steering these saliencies through entropy-based penalization and sample-wise cross-task alignment, Rep-MTL aims to mitigate negative transfer by maintaining the effective training of individual tasks instead pure conflict-solving, while explicitly promoting complementary information sharing. Experiments are conducted on four challenging MTL benchmarks covering both task-shift and domain-shift scenarios. The results show that Rep-MTL, even paired with the basic equal weighting policy, achieves competitive performance gains with favorable efficiency. Beyond standard performance metrics, Power Law exponent analysis demonstrates Rep-MTL's efficacy in balancing task-specific learning and cross-task sharing. The project page is available at HERE.

1. Introduction

Multi-Task Learning (MTL) [4] has garnered increasing attention in recent years, with notable success in computer vision [14, 23], natural language processing [3, 50], and other modalities [37, 38]. By leveraging multiple supervision signals at once, MTL models are expected to learn robust representation with reduced cost but better generalization compared to their single-task learning (STL) counterparts [20].

However, performance deterioration occurs as the tasks involved may not necessarily exhibit significant correlation, which induces conflicts in joint training [9, 66], *i.e.*, competing updates to the same architecture could impede the effective training of individual tasks, thus resulting in inferior convergence and worse generalization [22, 67]. This hence makes optimization an integral part of MTL, with its crux believed to be alleviating the *negative transfer* among tasks while exploiting their *positive complementarity* [57, 60, 64].

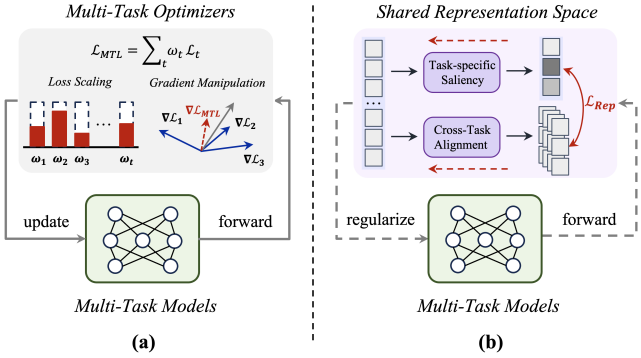

Figure 1. Overview of Rep-MTL and existing MTO methods. (a) Both loss scaling and gradient manipulation focus on optimizer-centric policies to address conflicts in model update. (b) Rep-MTL exploits shared representation space to facilitate cross-task sharing while preserving task-specific signals without optimizer changes.

As such, various studies have focused on addressing negative transfer through optimizer-centric loss scaling [33, 35] and gradient manipulation [2, 49]. Although more precise multi-task optimization (MTO) strategies have been introduced, their inconsistent efficacy has been increasingly recognized, especially in demanding scenarios [24, 27, 48, 64]. In parallel, the probe of task relationships has been extended into shared representation space [1, 21, 43], in which task saliencies are quantified by gradients w.r.t. shared representation [53], opening new avenues for MTO. In this paper, we argue that the representation space, where task interactions naturally occur, offers rich information and potential for operations beyond (and complementary to) optimizer designs, especially for facilitating task complementarity *explicitly*.

To this end, we introduce Rep-MTL, a representation-

centric approach that modulates multi-task training via task saliency (Sec. 3.1). As shown in Figure 2, it comprises two associative modules: (i) **T**ask-specific **S**aliency **R**egulation (TSR) that penalizes the task saliency distributions through an entropy-based regularization, ensuring that task-specific patterns can be reserved and remain distinctive during training, thereby mitigating negative transfer in MTO. (ii) **C**ross-task **S**aliency **A**lignment (CSA). Since prior methods rarely touched on this, we start with an intuitive question: *What additional message does representation space provide us during training?* – the rich sample-wise feature dimensions first come to our mind and inspire our design of CSA, which *explicitly* promotes inter-task complementarity by aligning sample-wise saliencies in a contrastive learning paradigm. Together, Rep-MTL as a regularization requires no further modifications to either optimizers or network architectures.

As aforementioned, MTO methods might perform worse in demanding settings. To rigorously validate Rep-MTL's effectiveness, we evaluate it from three key aspects: First, we conduct experiments on four challenging MTL benchmarks that encompass both task-shift and domain-shift scenarios, where most MTO baselines exhibit negative performance gains. Second, we examine Rep-MTL's robustness and practical applicability by analyzing its sensitivity to hyper-parameters (Sec. 4.5), learning rates (Appendix D.3), and its optimization speed (Sec. 4.6). Third, beyond standard performance metrics, we apply Power Law (PL) exponent analysis [44, 46] to validate if and how Rep-MTL influences model updates (Sec. 4.3). Specifically, it verifies how different trained model parts (*e.g.*, backbone or decoders) fit their related objectives (overall or task-specific losses) under different MTO methods. Overall, the empirical results show that: (i) Rep-MTL consistently achieves competitive performance, even with the basic equal weighting (EW). (ii) Rep-MTL is more efficient than most gradient manipulation methods (*e.g.*, $\sim$26% and $\sim$12% faster than Nash-MTL [49] and FairGrad [2], respectively). (iii) It successfully maintains effective training of individual tasks while exploiting inter-task complementarity for more robust MTL models.

Our contributions can thus be summarized as follows:

- We introduce Rep-MTL, a representation-centric MTO approach that aims to mitigate negative transfer while exploiting inter-task complementarity with task saliency.
- Rep-MTL as a regularization method complements existing MTO strategies, achieving competitive performance on diverse MTL benchmarks even with the basic EW.
- Observation and insights are obtained from empirical evidence: (i) Mitigating negative transfer could go beyond pure conflict-solving strategies. TSR offers another path by ensuring the effective training of individual tasks. (ii) The explicit cross-task sharing in CSA has shown significant potential, but remains under-explored in MTO.

2. Related Work

2.1. Multi-Task Optimization

Prior work in MTO operates under the assumption that the negative transfer stems from task conflicts in gradient directions and magnitude. Thus, three types of optimizer-centric methods have been developed: loss scaling, gradient manipulation, and hybrid ones. Below, we discuss where existing studies fall on each group and where our Rep-MTL stands.

Loss Scaling. Loss scaling methods optimize both shared and task-specific parameters by adjusting task-specific loss weights, evolving from EW [68] to more complex policies, such as homoscedastic uncertainty-based scaling [22], and loss changing-based adaptation [35]. GLS [10] minimizes the geometric mean loss, while IMTL-L [34] proposes an adaptive loss transformation to maintain constant weighted losses. RLW [28] employs probabilistic sampling from normal distributions, and IGBv2 [12] utilizes improbable gaps for scale computation. More recently, FAMO [33] balances different losses by decreasing them approximately at an equal rate. GO4Align [57] goes a step further through task grouping-based interaction for dynamic progress alignment.

Gradient Manipulation and Hybrid Methods. Gradient manipulation techniques adjust and aggregate task-specific gradients w.r.t. shared parameters to address conflicts. Early attempts focused on basic gradient modification: GradNorm [8] equalizes gradient magnitudes with learned task weights, whereas PCGrad [66] resolves conflicts by gradient projection. GradVac [63] and GradDrop [9] introduce the universal alignment and consistency-based gradient sign dropout, respectively. MGDA [13] for the first time views the MTO issue as a multi-objective optimization problem and pioneered the search for Pareto-optimal solutions. Subsequently, this paradigm was improved by CAGrad [32], which optimizes for the worst-case task improvement, and MoCo [15], which incorporates momentum estimation to eliminate bias. Nash-MTL [49] incorporates game theory to find Nash bargaining solutions, while IMTL-G [34] updates gradient directions that all their cosine similarities are equal. Recon [58] first employs Neural Architecture Search to deal with gradient conflicts. More recently, Aligned-MTL [55] leverages principal component alignment for stability. FairGrad [2] and STCH [31] reformulate MTO with utility maximization and smoothed Tchebycheff optimization. Hybrid methods [29, 30, 34, 36] combine the above two groups of technique to cash in their complementary strengths.

Shared Representation in MTO. Several studies [49, 54, 55] have utilized gradients w.r.t. shared representation rather than parameters to update models, which reduces the computational overhead with limited backpropagation through task-specific decoders. Among these, RotoGrad [21] and SRDML [1] stand as the closest studies

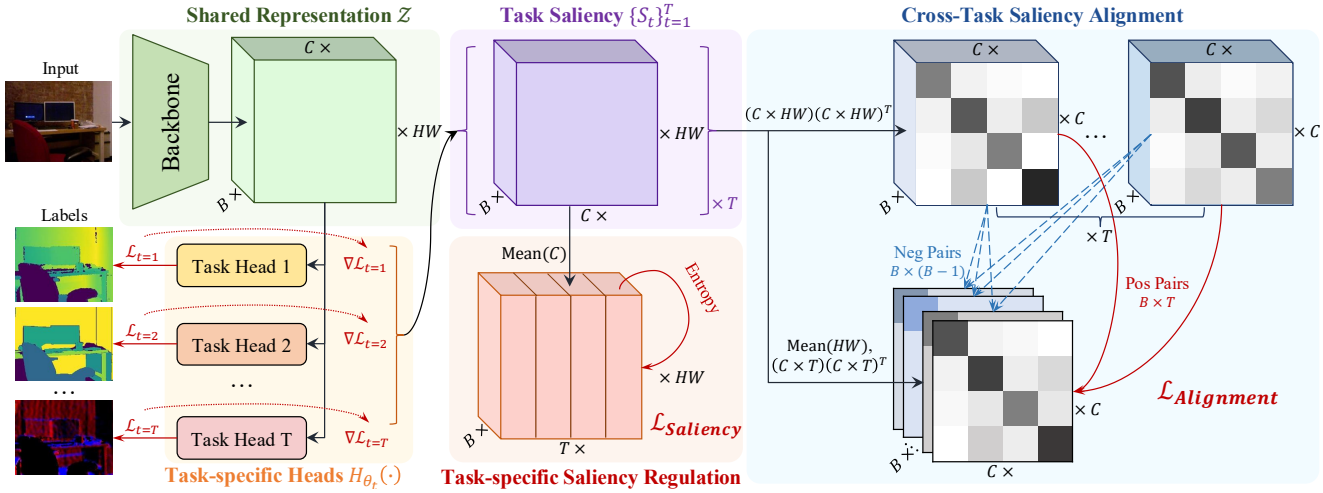

Figure 2. The Rep-MTL framework. It comprises two complementary task saliency driven modules: *(i) Task-specific Saliency Regulation (TSR)* that penalizes the saliency distribution to emphasize task-salient patterns during training, thereby mitigating negative transfer among tasks. *(ii) Cross-task Saliency Alignment (CSA)* that facilitates inter-task complementarity by aligning saliencies while maintaining task distinctiveness through positive and negative pairs. Together, Rep-MTL enables transfer across tasks while keeping task-specific patterns.

to Rep-MTL. RotoGrad employs feature rotation to minimize task semantic disparities, while SRDML learns task relations by regularizing task-wise similarity of saliencies. While they have made substantial progress, they solve task conflicts with task similarities and leave the essential inter-task complementarity entirely to the MTL architectures.

2.2. MTL Architectures

Hard parameter sharing (HPS) architecture [4], while fundamental for MTL, suffers from *negative transfer* among tasks. Subsequent work has thus focused on facilitating cross-task positive transfer through various architecture designs: from learnable fusion weights (*e.g.*, Cross-stitch [47], Sluice network [51]), to dynamic expert combinations (*e.g.*, MoE [56], MMoE [39], PLE [61]), and attention-based feature extraction like MTAN [35]. Moreover, several studies explore knowledge transfer at output layer through sequential modeling (ESSM [40]) or distillation (CrossDistil [65]). While these architectures have proven effective in exploiting inter-task complementarity, MTO has mainly focused on addressing negative transfer by solving gradient conflicts. In this work, we aim to bridge this gap by introducing an MTO method that *explicitly* facilitates such complementarity while maintaining the canonical HPS architecture.

3. Methodology

This section presents Rep-MTL, which advances multi-task learning through task saliency regularization in shared representation space. We first preview the MTO problem and representation-level task saliency (Sec. 3.1). Subsequently, we propose two complementary components of Rep-MTL: (i) task-specific saliency regulation that aims to mitigate negative MTL transfer by preserving task-salient learning patterns (Sec. 3.2), and (ii) sample-wise cross-task alignment for facilitating inter-task complementarity (Sec. 3.3).

3.1. Preliminaries and Task Saliency

Consider a set of T correlated tasks $\{\mathcal{T}_t\}_{t=1}^T$, where $T \in \mathbb{N}^+$ denotes the total number of tasks. In HPS architectures, model parameters $\theta = \{\theta_s, \{\theta_t\}_{t=1}^T\}$ comprise shared parameters θ_s for the backbone and task-specific parameters $\{\theta_t\}_{t=1}^T$ for decoders, where each θ_t corresponds to task $\mathcal{T}_t$.

For illustration, we present the framework in the context of computer vision. Given an input RGB image $X \in \mathbb{R}^{3\times H\times W}$, a shared encoder $E_{\theta_s}(\cdot)$ maps the input X into a latent feature space $Z = E_{\theta_s}(X) \in \mathbb{R}^{C\times H'\times W'}$, where C is channel dimension. Task-specific decoders $\{H_{\theta_t}(\cdot)\}_{t=1}^T$ then transform shared representation Z to diverse predictions $\{\hat{Y}_t\}_{t=1}^T$, where $\hat{Y}_t = H_{\theta_t}(Z)$. For each task $\mathcal{T}_t$, we derive the empirical loss function $\mathcal{L}_t(\theta_s, \theta_t)$. The conventional MTL learning objective can thus be formulated as:

$$\theta^* = \arg\min_{\theta} \sum_{t=1}^{T} \omega_t \mathcal{L}_t(\theta_s, \theta_t) \tag{1}$$

where $\omega_t > 0$ denotes task-specific weights adjusted by loss scaling methods. To prevent negative transfer, where optimizing one task deteriorates others due to the conflicting parameter updates, gradient manipulation policies harmonize parameter-wise gradients $\{\nabla_\theta \mathcal{L}_t\}_{t=1}^T$, especially those w.r.t. shared parameters θ_s during back-propagation:

$$g_t = \nabla_{\theta_s} \mathcal{L}_t(\theta_s, \theta_t), \quad \tilde{g} = h(\{g_t\}_{t=1}^T) \tag{2}$$

where g_t represents task-specific gradients and $\tilde{g}$ directly guides the updates of θ_s through specifically tailored transformation function $h(\cdot)$ in optimizers: $\theta_s^* = \theta_s - \eta \cdot \tilde{g}$.

As aforementioned, this work follow the representation-centric approach in MTO to first characterize *how different tasks* $\{\mathcal{T}_t\}_{t=1}^T$ *interact within the shared representation space* through task saliency $\mathcal{S}_t$, which could be quantified by its gradient w.r.t. shared representation Z as:

$$\mathcal{S}_t = \nabla_Z \mathcal{L}_t(\theta_s, \theta_t) \in \mathbb{R}^{B\times C\times H'\times W'} \tag{3}$$

where B denotes the training batch size. The feature-level $\{\mathcal{S}_t\}_{t=1}^T$ here quantifies how sensitively each task's objective responds to the changes within representations [43, 53]. Unlike parameter gradients for direct model updates, these saliencies serve as dynamic indicators of representation-level task dependencies, providing informative learning signals for identifying and even modulating task interactions.

More importantly, this representation-centric perspective enables us to proactively approach cross-task information sharing. By regulating the task saliency, we can identify features crucial for specific tasks (helping preserve task-specific patterns and thus mitigate negative transfer) and which features show consistent importance (facilitating complementarity). Note that prior work in MTO have mainly focused on resolving conflicts while neglecting the potential for *explicitly* leveraging inter-task complementarity, thus leaving this part to MTL architectures [35, 47, 65].

In the following sections, we present two complementary modules based on the above task saliency $\{\mathcal{S}_t\}_{t=1}^T$: (i) an entropy-based saliency regularization to preserve task-specific learning patterns, and (ii) a sample-wise contrastive alignment to *explicitly* exploit inter-task complementarity.

3.2. Task-specific Saliency Regulation

As aforementioned, the challenge lies in maintaining each task's effective training to prevent deterioration from negative transfer. To achieve this goal, we employ an entropy-based saliency regularization that preserves the task-specific learning patterns in representation space. As shown in Figure 2, instead of computing task similarity like most MTO methods, we begin by aggregating channel-wise saliencies in Eq. 3 to capture spatial patterns of task importance:

$$\hat{\mathcal{S}}_t = \frac{1}{|C|}\sum_c \mathcal{S}_{t,b,c,h,w} \in \mathbb{R}^{B\times H'\times W'} \tag{4}$$

where $\hat{\mathcal{S}}_t$ maintains the original spatial structure while reducing dimensionality for computational efficiency. Thus, $\hat{\mathcal{S}}_t$ captures how each spatial region $Z_{h,w}$ contributes to diverse task-specific learning processes. To further characterize these regions' relative importance across tasks, we normalize each $\hat{\mathcal{S}}_{i,t} \in \mathbb{R}^B$ into task-wise distribution $\mathcal{P}_{i,t}$:

$$\mathcal{P}_{i,t} = \frac{|\hat{\mathcal{S}}_{i,t}|}{\sum_{k=1}^T |\hat{\mathcal{S}}_{i,k}|} \tag{5}$$

This transforms raw spatial saliencies $\{\hat{\mathcal{S}}_{i,t}\}_{t=1}^T$ into interpretable probability distribution that identifies regions where specific tasks exhibit salient learning patterns, bridging task- and feature-level characteristics. Holistically, high probabilities indicate regions important to individual tasks, while uniform ones suggest rather common elements. Thus, to encourage an MTL model that keeps task-salient patterns during training, we introduce an entropy-based regulation:

$$\mathcal{L}_{tsr}(Z) = \frac{1}{BH'W'}\sum_{i=1}^{BH'W'}(-\sum_{t=1}^{T}\mathcal{P}_{i,t}\log\mathcal{P}_{i,t}) \tag{6}$$

This term encourages distinct task saliencies by penalizing high-entropy distributions. When spatial regions are task-salient (indicated by low entropy), the regularization $\mathcal{L}_{ts}$ encourages their task-specific patterns by penalizing excessive sharing caused by negative transfer. Empirical analysis in Sec. 4.3 and Figure 4 further demonstrate the improved task-specific learning quality and successful negative transfer mitigation of $\mathcal{L}_{ts}$. The rest of this section expands on the sample-wise contrastive alignment to further promote beneficial information sharing across tasks.

3.3. Sample-wise Cross-Task Saliency Alignment

While saliency distribution helps preserve task-specific dynamics, hitting optimal MTL performance requires leveraging common patterns across tasks. To address this complementary part, we present a contrastive alignment mechanism, as shown in Figure 2, that facilitates beneficial information sharing while maintaining task distinctiveness.

The core idea behind this is that similar patterns within task saliencies $\{\mathcal{S}_t\}_{t=1}^T$ indicate task-generic information that could benefit multiple tasks and should be consistently represented. To identify these shared patterns, we compute the affinity maps of saliency $\mathcal{S}_t$ for task $\mathcal{T}_t$:

$$\mathcal{M}_t = \mathcal{S}_t\mathcal{S}_t^\top \in \mathbb{R}^{B\times C\times C}, \tag{7}$$

where $\mathcal{S}_t$ indicates the saliency in Eq. 3. These affinity maps capture the mutual influence patterns between features in each task's optimization, encoding how different channels interact during training. Drawing inspiration from contrastive learning [6, 7, 16, 18], we compute the reference anchor $\mathcal{A}_b$ for each sample $b \in [B]$, which serves as candidates for information sharing in subsequent alignment:

$$\mathcal{A}_b = \frac{1}{H'W'}\sum_{hw}\mathcal{S}_{hw,b}, \quad \hat{\mathcal{A}}_b = \mathcal{A}_b\mathcal{A}_b^\top \in \mathbb{R}^{C\times C} \tag{8}$$

where $\mathcal{S}_{hw,b}$ denotes saliency maps for sample b. These anchors serve as stable points for cross-task alignment, capturing potential shared dynamics that emerge across diverse tasks for identical samples. We then L_2-norm both anchors $\hat{\mathcal{A}}_b$ and affinities $\mathcal{M}_t$ to obtain z_b^a and z_b^t, ensuring scale-invariant comparisons. For anchors z_b^a, we treat the related

Table 1. Performance on NYUv2 [59] dataset (3 indoor scene understanding tasks) with DeepLabV3+ [5] network architecture. ↑ (↓) indicates higher (lower) metric values are better. The best and second-base results for each metric are highlighted in **bold** and underline, respectively. ‡ indicates results from our implementation using `LibMTL` [27] codebase, which provides official support for related methods.

	Segmentation		Depth Estimation		Surface Normal Prediction						
					Angle Dist.		Within $t°$				
Method	mIoU↑	Pix. Acc.↑	Abs. Err↓	Rel. Err↓	Mean↓	Median↓	11.25↑	22.5↑	30↑	$\Delta\mathbf{p}_{\text{task}}$↑	$\Delta\mathbf{p}_{\text{metric}}$↑
Single-Task Baseline	53.50	75.39	0.3926	0.1605	21.99	**15.16**	**39.04**	**65.00**	75.16	0.00	0.00
EW	53.93	75.53	0.3825	0.1577	23.57	17.01	35.04	60.99	72.05	$-1.78_{\pm 0.45}$	−3.85
GLS [10]	54.59	**76.06**	0.3785	0.1555	22.71	16.07	36.89	63.11	73.81	$+0.30_{\pm 0.30}$	−1.10
RLW [28]	54.04	75.58	0.3827	0.1588	23.07	16.49	36.12	62.08	72.94	$-1.10_{\pm 0.40}$	−2.64
UW [22]	54.29	75.64	0.3815	0.1583	23.48	16.92	35.26	61.17	72.21	$-1.52_{\pm 0.39}$	−3.54
DWA [35]	54.06	75.64	0.3820	0.1564	23.70	17.11	34.90	60.74	71.81	$-1.71_{\pm 0.25}$	−3.96
IMTL-L [34]	53.89	75.54	0.3834	0.1591	23.54	16.98	35.09	61.06	72.12	$-1.92_{\pm 0.25}$	−3.90
IGBv2 [12]	**54.61**	76.00	0.3817	0.1576	22.68	15.98	37.14	63.25	73.87	$+0.05_{\pm 0.29}$	−1.15
MGDA [13]	53.52	74.76	0.3852	0.1566	22.74	16.00	37.12	63.22	73.84	$-0.64_{\pm 0.25}$	−1.65
GradNorm [8]	53.91	75.38	0.3842	0.1571	23.17	16.62	35.80	61.90	72.84	$-1.24_{\pm 0.15}$	−2.90
PCGrad [66]	53.94	75.62	0.3804	0.1578	23.52	16.93	35.19	61.17	72.19	$-1.57_{\pm 0.44}$	−3.60
GradDrop [9]	53.73	75.54	0.3837	0.1580	23.54	16.96	35.17	61.06	72.07	$-1.85_{\pm 0.39}$	−3.84
GradVac [63]	54.21	75.67	0.3859	0.1583	23.58	16.91	35.34	61.15	72.10	$-1.75_{\pm 0.39}$	−3.72
IMTL-G [34]	53.01	75.04	0.3888	0.1603	23.08	16.43	36.24	62.23	73.06	$-1.89_{\pm 0.54}$	−3.09
CAGrad [32]	53.97	75.54	0.3885	0.1588	22.47	15.71	37.77	63.82	74.30	$-0.27_{\pm 0.35}$	−0.98
MTAdam [42]	52.67	74.86	0.3873	0.1583	23.26	16.55	36.00	61.92	72.74	$-1.97_{\pm 0.23}$	−3.36
Nash-MTL [49]	53.41	74.95	0.3867	0.1612	22.57	15.94	37.30	63.40	74.09	$-1.01_{\pm 0.13}$	−1.76
MetaBalance [19]	53.92	75.57	0.3901	0.1594	22.85	16.16	36.72	62.91	73.62	$-1.06_{\pm 0.17}$	−2.15
MoCo [15]	52.25	74.56	0.3920	0.1622	22.82	16.24	36.58	62.72	73.49	$-2.25_{\pm 0.51}$	−3.03
Aligned-MTL [55]	52.94	75.00	0.3884	0.1570	22.65	16.07	36.88	63.18	73.94	$-0.98_{\pm 0.56}$	−1.92
FairGrad‡ [2]	53.01	75.14	0.3795	0.1573	22.51	16.02	36.93	63.39	74.17	$-0.47_{\pm 0.56}$	−1.46
STCH‡ [31]	53.86	75.49	0.3759	0.1547	22.69	16.17	36.70	62.96	73.82	$+0.06_{\pm 0.11}$	−1.35
IMTL [34]	53.63	75.44	0.3868	0.1592	22.58	15.85	37.44	63.52	74.09	$-0.57_{\pm 0.24}$	−1.38
DB-MTL [30]	53.92	75.60	0.3768	0.1557	21.97	15.37	38.43	64.81	**75.24**	$+1.15_{\pm 0.16}$	+0.56
Rep-MTL (EW)	54.59	76.04	**0.3750**	**0.1542**	**21.91**	15.28	38.37	64.72	75.05	**+1.70**$_{\pm 0.29}$	**+0.95**

task affinities z_b^t from identical sample as positive pairs, while those of different samples within a batch serve as negative pairs. As such, cross-task alignment is formulated as:

$$\mathcal{L}_{csa} = \frac{1}{B}\sum_{b=1}^{B} -\log \frac{\exp(\text{sim}(z_b^a, z_b^t)/\tau)}{\sum_{k\neq b}\exp(\text{sim}(z_b^a, z_k^a)/\tau)}, \quad (9)$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ controls the concentration of positive pairs (z_b^a, z_b^t) relative to the negative ones (z_b^a, z_k^a). As such, the balance between positive and negative pairs as well as the inclusion of batch information maintains task distinctiveness while promoting information sharing across tasks. Empirical analysis in Sec. 4.3 and Figure 3 shows that this contrastive saliency alignment facilitates models' optimization towards the overall MTL objectives, indicating that the intrinsic inter-task complementarity has been effectively excavated by CSA $\mathcal{L}_{csa}$.

3.4. Joint Optimization

To achieve robust multi-task training that both preserves task-specific learning signals and enables cross-task sharing, we combine MTL objectives with above regularization:

$$\mathcal{L}_{\text{Rep}} = \sum_{t=1}^{T} \mathcal{L}_t(\theta_s, \theta_t) + \lambda_{tsr}\mathcal{L}_{tsr}(Z) + \lambda_{csa}\mathcal{L}_{csa}(Z), \quad (10)$$

where λ_1 and λ_2 balance the influence of above two regularization terms. During joint optimization, gradients from all these components flow through the multi-task model, introducing implicit adjustment to model parameter updates.

4. Experiment

We evaluate Rep-MTL on four MTO benchmarks covering both task-shift and domain-shift scenarios: (i) NYUv2 [59] (3-task indoor scene understanding), (ii) Cityscapes [11] (2-task urban scene understanding), (iii) Office-31 [52] (3-domain image classification), and (iv) Office-Home [62] (4-domain image classification). To ensure thorough and rigorous evaluation, we follow the more challenging benchmark settings [30] for NYUv2 and Cityscapes, where most baselines exhibit negative performance gains (see Table 1, 2).

In this section, we first introduce the baselines and evaluation metrics, followed by quantitative results on scene understanding (Sec. 4.1) and image classification (Sec. 4.2).

Table 2. Performance on Cityscapes [11] dataset (2 scene understanding tasks) with DeepLabV3+ [5] architecture. ↑ (↓) indicates higher (lower) metric values are better. The best and second-best results are marked in **bold** and underline, respectively. ‡ indicates results from our implementation using `LibMTL` [27] codebase.

	Segmentation		Depth Estimation		Δp_{task}↑
Method	mIoU↑	Pix. Acc.↑	Abs. Err↓	Rel. Err↓	
Single-Task Baseline	69.06	91.54	0.01282	43.53	0.00
EW	68.93	91.58	0.01315	45.90	$-2.05_{\pm 0.56}$
GLS [10]	68.69	91.45	0.01280	44.13	$-0.39_{\pm 1.06}$
RLW [28]	69.03	91.57	0.01343	44.77	$-1.91_{\pm 0.21}$
UW [22]	69.03	91.61	0.01338	45.89	$-2.45_{\pm 0.68}$
DWA [35]	68.97	91.58	0.01350	45.10	$-2.24_{\pm 0.28}$
IMTL-L [34]	68.98	91.59	0.01340	45.32	$-2.15_{\pm 0.88}$
IGBv2 [12]	68.44	91.31	0.01290	45.03	$-1.31_{\pm 0.61}$
MGDA [13]	69.05	91.53	0.01280	44.07	$-0.19_{\pm 0.30}$
GradNorm [8]	68.97	91.60	0.01320	44.88	$-1.55_{\pm 0.70}$
PCGrad [66]	68.95	91.58	0.01342	45.54	$-2.36_{\pm 1.17}$
GradDrop [9]	68.85	91.54	0.01354	44.49	$-2.02_{\pm 0.74}$
GradVac [63]	68.98	91.58	0.01322	46.43	$-2.45_{\pm 0.54}$
IMTL-G [34]	69.04	91.54	0.01280	44.30	$-0.46_{\pm 0.67}$
CAGrad [32]	68.95	91.60	0.01281	45.04	$-0.87_{\pm 0.88}$
MTAdam [42]	68.43	91.26	0.01340	45.62	$-2.74_{\pm 0.20}$
Nash-MTL [49]	68.88	91.52	**0.01265**	45.92	$-1.11_{\pm 0.21}$
MetaBalance [19]	69.02	91.56	0.01270	45.91	$-1.18_{\pm 0.58}$
MoCo [15]	69.62	91.76	0.01360	45.50	$-2.40_{\pm 1.50}$
Aligned-MTL [55]	69.00	91.59	0.01270	44.54	$-0.43_{\pm 0.44}$
FairGrad‡ [2]	68.84	91.48	0.01270	46.26	$-1.43_{\pm 0.63}$
STCH‡ [31]	68.21	91.23	0.01280	**43.17**	$-0.15_{\pm 0.38}$
IMTL [34]	69.07	91.55	0.01280	44.06	$-0.32_{\pm 0.10}$
DB-MTL [30]	69.17	91.56	0.01280	43.46	$+0.20_{\pm 0.40}$
Rep-MTL (EW)	**69.72**	**91.85**	0.01270	43.42	**+0.62**$_{\pm 0.53}$

We then present PL exponent analysis (Sec. 4.3) with ablation studies (Sec. 4.4) that validate Rep-MTL's effectiveness beyond benchmarking results. There are also evaluations to assess Rep-MTL's robustness and applicability, including sensitivity of hyperparameters (Sec. 4.5) and learning rates (Appendix D.3), and also its optimization speed (Sec. 4.6).

Baselines. We compare our Rep-MTL against 23 popular MTO algorithms, including loss scaling policies (GLS [10], RLW [28], UW [22], DWA [35], IGBv2 [12]), gradient manipulation (MGDA [13], GradNorm [8], PCGrad [66], GradDrop [9], GradVac [63], CAGrad [32], MTAdam [42], Nash-MTL [49], MetaBalance [19], MoCo [15], Aligned-MTL [55]), and hybrid approaches (IMTL [34], DB-MTL [30]). We also implement recent FairGrad [2], and STCH [31] on NYUv2 [59] and Cityscapes [11] using the open-source `LibMTL` [27] codebase. For all included algorithms, HPS architecture is employed for fair comparison.

Evaluation Metrics. Task evaluation follows established metrics [30, 35, 55]: For semantic segmentation, we use mean Intersection over Union (mIoU) and pixel-wise accuracy (Pix Acc). Depth estimation performance is measured using relative error (Rel Err) and absolute error (Abs Err). Surface normal prediction is evaluated with mean and median angle errors, along with percentage of normals within angular thresholds of t° ($t = 11.25, 22.5, 30$). We quantify MTL performance improvements relative to STL baselines as average gains over tasks Δp_{task} and metrics Δp_{metric}:

$$\Delta p_{metric} = \frac{1}{T}\sum_{t=1}^{T}(-1)^{\sigma_t}\frac{(M_{m,t} - M_{b,t})}{M_{b,t}}, \tag{11}$$

$$\Delta p_{task} = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n_t}(-1)^{\sigma_{tk}}\frac{(M_{m,tk} - M_{b,tk})}{M_{b,tk}} \tag{12}$$

where T denotes the total number of tasks, n_t represents the number of evaluation metrics for task t, and $M_{m,t}$ indicates the performance of method m on task t. The sign coefficient σ_t (σ_{tk}) equals 0 for positive metrics (*e.g.*, mIoU, Pix Acc) and 1 for negative metrics (*e.g.*, Err, Dist.), ensuring consistent interpretation. Each experiment is repeated three times. Please refer to Appendix A for experimental settings.

4.1. Scene Understanding Tasks

Datasets and Settings. We evaluate our Rep-MTL on two scene understanding benchmarks: (i) NYUv2 [59] comprises three tasks (13-class semantic segmentation, depth estimation, and surface normal prediction), with 795 training and 654 validation images; (ii) Cityscapes [11] owns two tasks (7-class semantic segmentation and depth estimation) with 2975 training and 500 testing images. Following previous settings [30], we employ DeepLabV3+ [5] as our architecture, where a dilated ResNet-50 [17] as the encoder and ASPP as decoders. Please view Appendix A for details.

NYUv2 Results. Table 1 shows that Rep-MTL achieves competitive multi-task performance gains on the NYUv2 benchmark, as measured by Δp_{task} and Δp_{metric}. **(i) EW baseline comparison:** Compared to the gray-marked EW baseline, Rep-MTL shows significant improvements, with gains of $+3.48$ in Δp_{task} and $+4.8$ in Δp_{metric}, which improves performance the most without extra optimizer modifications. **(ii) SOTA comparison:** Rep-MTL outperforms previous leading methods, DB-MTL [30], by about 48% in Δp_{task} ($+1.70$ *vs.* $+1.15$) and nearly $\sim$ 70% in Δp_{metric} ($+0.95$ *vs.* $+0.56$). Furthermore, Rep-MTL surpasses DB-MTL in $6/9$ sub-task metrics, showcasing its effectiveness.

Cityscapes Results. Table 2 demonstrates Rep-MTL's effectiveness on the Cityscapes benchmark, where it achieves the best results in semantic segmentation and the second-best results in two depth estimation metrics. **(i) EW baseline comparison:** Rep-MTL improves upon EW baseline by $+2.67$ in Δp_{task}, showcasing its ability to improve MTL performance across datasets. **(ii) SOTA comparison:** DB-MTL [30] is the only method that surpasses STL baseline on this challenging benchmark setting. However, Rep-MTL slightly exceeds the powerful DB-MTL in Δp_{task} ($+0.62$ *vs.* $+0.20$), which indicates an improvement of multi-task dense prediction in outdoor scene understanding scenarios.

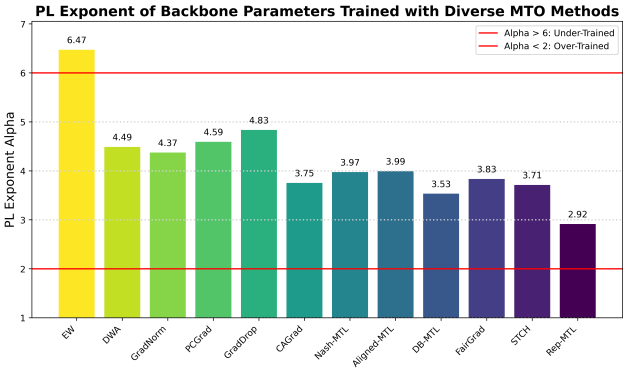

Figure 3. Comparison of PL exponent alpha [46] for backbone parameters trained with diverse MTO methods on NYUv2 [59]. It validates how well the backbone adapts to MTL objectives, where lower values indicate more effective training. Values outside $[2, 4]$ suggest potential over- or under-training. We leverage this to show how methods affect model updates, as well-trained backbones suggest beneficial cross-task sharing to the overall MTL objectives.

4.2. Image Classification Tasks

Datasets and Settings. To further validate Rep-MTL's effectiveness in domain-shift scenarios, the following datasets are included: (i) Office-31 [52] which contains 4110 images from three domains (tasks): Amazon, DSLR, and Webcam. Each task has 31 object categories. (ii) Office-Home [62] which contains 15500 images from four domains (tasks): artistic, clipart, product, and real-world. Each of them has 65 object categories in office and home settings. We use the data split as 60% for training, 20% for validation, and 20% for testing. Following Lin et al. [30], we use ResNet18 as network architecture. Please view Appendix A for details.

Results. As shown in Table 3, Rep-MTL still maintains its effectiveness in domain-shift scenarios [62]. Concretely, Rep-MTL surpasses EW by $+0.97$ in average performance and achieves a positive Δp_{task} of $+0.41$, which advances the previous SOTA by approximately 140% ($+0.41$ *vs.* $+0.17$). This validates Rep-MTL's capability to exploit inter-task complementarity even if there are domain gaps. Situations on Office-31 [52] (Appendix B) are even more challenging, where Rep-MTL exhibits the best results on Webcam, average performance (Avg.), and Δp_{task} improvement, slightly outperforming previous SOTA by 25% ($+1.31$ *vs.* $+1.05$).

4.3. Power Law (PL) Exponent Analysis

We analyze Power Law (PL) exponent alpha [44, 46] derived from Heavy-Tailed Self-Regularization (HT-SR) theory [41, 45]. For each layer's weight matrix W, the PL exponent is computed by fitting the Empirical Spectral Density of $X = W^T W$ to $\rho(\lambda) \sim \lambda^{-\alpha}$, where λ are eigenvalues of the correlation matrix. This scale-invariant metric has proven effective in evaluating training quality without access to training or test data [25, 26], making it particularly suitable for analyzing MTL models: α quantifies how well different model parts (*e.g.*, shared backbone or task-specific decoders) adapt to their corresponding objectives (*e.g.*, the overall MTL loss or task-specific losses). Thus, by examining α values of different components of MTL model parameters, we can validate both the MTO methods' ability to assist the training process, *i.e.*, mitigating negative transfer among tasks while exploiting inter-task complementarity. In practice, well-trained models typically exhibit $\alpha \in [2, 4]$, while poorly-trained or over-parameterized models tend to show $\alpha \gg 4$, providing an applicable tool for assessing the overall effectiveness of MTO methods. In particular, a low α value for backbone often indicates effective cross-task information sharing for the overall MTL objectives, while low and balanced values across heads suggest high-quality training of individual tasks, thus indicating less negative transfer.

In this work, we analyze DeepLabV3+ models trained with different MTO methods on NYUv2 [59]. As shown in Figure 3, models trained with Rep-MTL exhibit superior $\alpha = 2.92$ for shared parameters θ_s compared to other MTO methods, which indicates more favorable inter-task complementarity. Moreover, Figure 4 reveals that Rep-MTL yields lower and more balanced PL exponent $(2.51, 2.46, 2.53)$ for decoders, demonstrating effective training of individual tasks, thereby mitigating negative transfer. Please view Appendix D for more details. We also incorporate PL exponent analysis into ablation studies in Sec. 4.4 to explore how the TSR and CSA modules separately influence model training.

Table 3. Performance on Office-Home [62] dataset with 4 diverse image classification tasks. ↑ indicates the higher the metric values, the better the methods' performance. The best and second-best results of each metric are marked in **bold** and underline, respectively.

Method	Artistic	Clipart	Product	Real	Avg.↑	Δp_{task}↑
Single-Task Baseline	65.59	**79.60**	**90.47**	80.00	78.91	0.00
EW	65.34	78.04	89.80	79.50	$78.17_{\pm 0.37}$	$-0.92_{\pm 0.59}$
GLS [10]	64.51	76.85	89.83	79.56	$77.69_{\pm 0.27}$	$-1.58_{\pm 0.46}$
RLW [28]	64.96	78.19	89.48	**80.11**	$78.18_{\pm 0.12}$	$-0.92_{\pm 0.14}$
UW [22]	65.97	77.65	89.41	79.28	$78.08_{\pm 0.30}$	$-0.98_{\pm 0.46}$
DWA [35]	65.27	77.64	89.05	79.56	$77.88_{\pm 0.28}$	$-1.26_{\pm 0.49}$
IMTL-L [34]	65.90	77.28	89.37	79.38	$77.98_{\pm 0.38}$	$-1.10_{\pm 0.61}$
IGBv2 [12]	65.59	77.57	89.79	78.73	$77.92_{\pm 0.21}$	$-1.21_{\pm 0.22}$
MGDA [13]	64.19	77.60	89.58	79.31	$77.67_{\pm 0.20}$	$-1.61_{\pm 0.34}$
GradNorm [8]	66.28	77.86	88.66	79.60	$78.10_{\pm 0.63}$	$-0.90_{\pm 0.93}$
PCGrad [66]	66.35	77.18	88.95	79.50	$77.99_{\pm 0.19}$	$-1.04_{\pm 0.32}$
GradDrop [9]	63.57	77.86	89.23	79.35	$77.50_{\pm 0.23}$	$-1.86_{\pm 0.24}$
GradVac [63]	65.21	77.43	89.23	78.95	$77.71_{\pm 0.19}$	$-1.49_{\pm 0.28}$
IMTL-G [34]	64.70	77.17	89.61	79.45	$77.98_{\pm 0.38}$	$-1.10_{\pm 0.61}$
CAGrad [32]	64.01	77.50	89.65	79.53	$77.73_{\pm 0.16}$	$-1.50_{\pm 0.29}$
MTAdam [42]	62.23	77.86	88.73	77.94	$76.69_{\pm 0.65}$	$-2.94_{\pm 0.85}$
Nash-MTL [49]	66.29	78.76	90.04	**80.11**	$78.80_{\pm 0.52}$	$-0.08_{\pm 0.69}$
MetaBalance [19]	64.01	77.50	89.72	79.24	$77.61_{\pm 0.42}$	$-1.70_{\pm 0.54}$
MoCo [15]	63.38	79.41	90.25	78.70	77.93	-
Aligned-MTL [55]	64.33	76.96	89.87	79.93	$77.77_{\pm 0.70}$	$-1.50_{\pm 0.89}$
IMTL [34]	64.07	76.85	89.65	79.81	$77.59_{\pm 0.29}$	$-1.72_{\pm 0.45}$
DB-MTL [30]	**67.42**	77.89	90.43	80.07	$78.95_{\pm 0.35}$	$+0.17_{\pm 0.44}$
Rep-MTL (EW)	67.40	78.75	90.37	80.04	$\mathbf{79.14}_{\pm 0.41}$	$\mathbf{+0.41}_{\pm 0.58}$

4.4. Ablation Study

The proposed Rep-MTL includes two key components: (i) TSR optimizing individual task learning via entropy-based

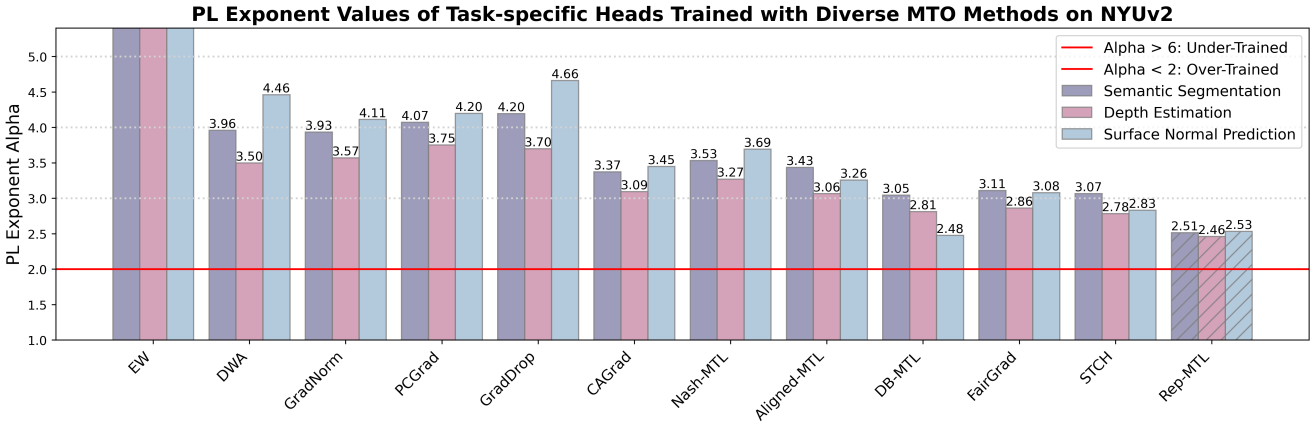

Figure 4. Visualization of PL exponent alpha [44, 46] for tasks-specific heads (3 tasks) trained with different MTO methods on NYUv2 [59]. PL exponent quantifies how well each decoder adapts to its task-specific objective, where lower values practically indicate more effective model training. Values outside the range $[2, 6]$ suggest potential over- or under-training due to task conflicts. The variation across decoders of each method indicates training imbalance. We employ this to evaluate how diverse MTO approaches influence decoder parameter updates, as well-trained decoders should exhibit both *low and balanced* values, indicating effective individual task training with successful negative transfer mitigation. For display, we constrain α range to $(0, 6)$ though EW results $(10.26, 7.01, 14.25)$ extend beyond this range.

regularization, and (ii) CSA which facilitates inter-task information sharing. Thus, to validate the contribution of each component, we conduct ablation studies using standard performance metrics and PL exponent analysis on NYUv2 [59] and Cityscapes [11] with results in Table 4 and Appendix C.

The CSA Module. Table 4 examines four configurations: (i) a baseline using neither component; (ii) CSA only; (iii) TSR only; and (iv) complete Rep-MTL with both components. The results demonstrate that both TSR and CSA individually contribute to positive Δp_{task}, while their combination yields the best results, showcasing their effectiveness. Notably, CSA alone yields greater performance gains than TSR alone, which is corroborated by our PL exponent analysis (Appendix C.3). It shows that even without TSR, CSA helps maintain balanced PL exponents across tasks-specific heads relative to baselines, suggesting that CSA effectively enables the model to leverage inter-task complementarity for joint training. This also shows the importance of *explicit designs for exploiting inter-task complementarity* in MTO.

The TSR Module. Beyond standard performance metrics, PL exponent analysis provides deeper insights into how each component functions. As shown in Appendix C.2, CSA leads to lower backbone PL exponent α within optimal range, indicating more effective task-generic feature learning from shared information. Meanwhile, TSR yields lower and more balanced PL exponents across task-specific decoders (Appendix C.3), demonstrating its capability in maintaining effective training of individual tasks, thereby mitigating negative transfer. It shows that *tackling negative transfer could go beyond pure conflict-resolving designs*. Approaches like TSR, which aim to emphasize individual tasks' training, may represent a promising and complementary direction. Together, these findings provide compelling evidence for both TSR and CSA's effectiveness.

4.5. Hyperparameter Sensitivity

For practical deployment, robust algorithms are expected to maintain stable performance across a reasonable range of hyperparameter settings. This directly impacts the method's applicability, especially in resource-constrained scenarios.

Table 4. Ablation study of Rep-MTL comprising two complementary components on NYUv2 [59] and Cityscapes [11] in terms of task-level performance gains $\Delta p_{task} \uparrow$ relative to STL baselines.

Cross-task Saliency Alignment (CSA)	Task-specific Saliency Regulation (TSR)	NYUv2 $\Delta p_{task} \uparrow$	Cityscapes $\Delta p_{task} \uparrow$
✗	✗	$-1.78_{\pm 0.45}$	$-2.05_{\pm 0.56}$
✓	✗	$+1.06_{\pm 0.27}$	$+0.21_{\pm 0.68}$
✗	✓	$+0.23_{\pm 0.29}$	$-0.34_{\pm 0.24}$
✓	✓	$\mathbf{+1.70}_{\pm 0.29}$	$\mathbf{+0.62}_{\pm 0.53}$

Empirical analysis in Appendix D.1 verifies that our Rep-MTL consistently achieves positive gains ($\Delta p_{task} > 0$) across a wide range of weighting coefficients ($\lambda_{tsr}, \lambda_{csa} \in [0.7, 1.5]$), which reduces the need for meticulous hyperparameter tuning. We also analyze Rep-MTL's learning rate sensitivity in Appendix D.3 to further show its robustness.

4.6. Efficiency Analysis

Optimization speed remains crucial for MTL. Thus, we conduct empirical analysis on NYUv2 [59] in Appendix D.2. while Rep-MTL requires increased runtime than loss scaling due to gradient computation, it exhibits better efficiency compared to most gradient manipulation methods approximately 26% faster than Nash-MTL [49] and 12% faster than FairGrad [2]), while delivering superior performance gains.

5. Conclusion

This paper presents Rep-MTL, a regularization-based MTO approach that leverages representation-level task saliency to advance multi-task training. By operating directly on task representation space, Rep-MTL aims to preserve the effective training of individual tasks while explicitly exploiting inter-task complementarity. Experiments not only reveal Rep-MTL's competitive performance, but highlight the significant yet largely untapped potential of directly regularizing representation space for more effective MTL systems.

References

[1] Guangji Bai and Liang Zhao. Saliency-regularized deep multi-task learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 15–25, 2022. 1, 2

[2] Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *Forty-first International Conference on Machine Learning*, 2024. 1, 2, 5, 6, 8

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 1

[4] Rich Caruana. Multitask learning. *Machine Learning*, 28(1): 41–75, 1997. 1, 3

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 5, 6, 12

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 4

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 4

[8] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, 2018. 2, 5, 6, 7, 13

[9] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *Neural Information Processing Systems*, 2020. 1, 2, 5, 6, 7, 13

[10] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir A Rawashdeh. MultiNet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2, 5, 6, 7, 13

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 5, 6, 8, 12

[12] Yanqi Dai, Nanyi Fei, and Zhiwu Lu. Improvable gap balancing for multi-task learning. In *Uncertainty in Artificial Intelligence*, pages 496–506. PMLR, 2023. 2, 5, 6, 7, 13

[13] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012. 2, 5, 6, 7, 13

[14] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2051–2060, 2017. 1

[15] Heshan Devaka Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *International Conference on Learning Representations*, 2023. 2, 5, 6, 7, 13

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 12

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 4

[19] Yun He, Xue Feng, Cheng Cheng, Geng Ji, Yunsong Guo, and James Caverlee. MetaBalance: improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *ACM Web Conference*, 2022. 5, 6, 7, 13

[20] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*, 2024. 1

[21] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2021. 1, 2

[22] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 5, 6, 7, 13

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[24] Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan K Mudigonda. In defense of the unitary scalarization for deep multi-task learning. *Advances in*

Neural Information Processing Systems, 35:12169–12183, 2022. 1

[25] Siyuan Li, Juanxi Tian, Zedong Wang, Luyuan Zhang, Zicheng Liu, Weiyang Jin, Yang Liu, Baigui Sun, and Stan Z Li. Unveiling the backbone-optimizer coupling bias in visual representation learning. *arXiv preprint arXiv:2410.06373*, 2024. 7

[26] Siyuan Li, Zedong Wang, Zicheng Liu, Juanxi Tian, Di Wu, Cheng Tan, Weiyang Jin, and Stan Z. Li. Openmixup: Open mixup toolbox and benchmark for visual representation learning, 2024. 7

[27] Baijiong Lin and Yu Zhang. Libmtl: A python library for deep multi-task learning. *The Journal of Machine Learning Research*, 24(1):9999–10005, 2023. 1, 5, 6

[28] Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. 2, 5, 6, 7, 12, 13

[29] Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. 2

[30] Baijiong Lin, Weisen Jiang, Feiyang Ye, Yu Zhang, Pengguang Chen, Ying-Cong Chen, Shu Liu, and James Kwok. Dual-balancing for multi-task learning, 2024. 2, 5, 6, 7, 12, 13

[31] Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. In *Forty-first International Conference on Machine Learning*, 2024. 2, 5, 6

[32] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Neural Information Processing Systems*, 2021. 2, 5, 6, 7, 13

[33] Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[34] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021. 2, 5, 6, 7, 12, 13

[35] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019. 1, 2, 3, 4, 5, 6, 7, 13

[36] Shikun Liu, Stephen James, Andrew Davison, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *Transactions on Machine Learning Research*, 2022. 2

[37] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2022. 1

[38] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024. 1

[39] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939, 2018. 3

[40] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1137–1140, 2018. 3

[41] Michael Mahoney and Charles Martin. Traditional and heavy tailed self regularization in neural network models. In *International Conference on Machine Learning*, pages 4284–4293. PMLR, 2019. 7, 14

[42] Itzik Malkiel and Lior Wolf. MTAdam: Automatic balancing of multiple training loss terms. In *Conference on Empirical Methods in Natural Language Processing*, 2021. 5, 6, 7, 13

[43] Dayou Mao, Yuhao Chen, Yifan Wu, Maximilian Gilles, and Alexander Wong. Robust analysis of multi-task learning on a complex vision system. *arXiv preprint arXiv:2402.03557*, 2024. 1, 4

[44] Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021. 2, 7, 8, 12, 14

[45] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021. 7, 14

[46] Charles H Martin, Tongsu Peng, and Michael W Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):4122, 2021. 2, 7, 8, 12, 13, 14

[47] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4

[48] David Mueller, Mark Dredze, and Nicholas Andrews. Can optimization trajectories explain multi-task transfer? *arXiv preprint arXiv:2408.14677*, 2024. 1

[49] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pages 16428–16446. PMLR, 2022. 1, 2, 5, 6, 7, 8, 12, 13

[50] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[51] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4822–4829, 2019. 3

[52] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 2010. 5, 7, 12, 13

[53] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 4

[54] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 2

[55] Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 5, 6, 7, 13

[56] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 3

[57] Jiayi Shen, Cheems Wang, Zehao Xiao, Nanne Van Noord, and Marcel Worring. Go4align: Group optimization for multi-task alignment. *arXiv preprint arXiv:2404.06486*, 2024. 1, 2

[58] Guangyuan Shi, Qimai Li, Wenlong Zhang, Jiaxin Chen, and Xiao-Ming Wu. Recon: Reducing conflicting gradients from the root for multi-task learning. *ArXiv*, abs/2302.11289, 2023. 2

[59] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, 2012. 5, 6, 7, 8, 12, 13, 14, 15, 16

[60] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, pages 9120–9132. PMLR, 2020. 1

[61] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 269–278, 2020. 3

[62] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 7, 12

[63] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021. 2, 5, 6, 7, 13

[64] Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. Do current multi-task optimization methods in deep learning even help? *Advances in neural information processing systems*, 35:13597–13609, 2022. 1, 15

[65] Chenxiao Yang, Junwei Pan, Xiaofeng Gao, Tingyu Jiang, Dapeng Liu, and Guihai Chen. Cross-task knowledge distillation in multi-task recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4318–4326, 2022. 3, 4

[66] 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*, 33:5824–5836, 2020. 1, 2, 5, 6, 7, 13

[67] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022. 1

[68] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34 (12):5586–5609, 2022. 2

Appendix

This appendix offers additional empirical analyses, experimental results, and further discussions of our work. The appendix sections are organized as follows:

- In Appendix A, we provide experimental setups and implementation specifications across all four benchmarks in this paper, including NYUv2 [59], Cityscapes [11], Office-Home [62], and Office-31 [52]. This includes comprehensive information on employed network architectures, optimization algorithms, training protocols, loss functions, and hyper-parameter configurations.
- In Appendix B, we provide complete experimental results on the Office-31 dataset, which were omitted from the main manuscript due to space constraints. We also discuss the proposed Rep-MTL method combined with all experimental results from four benchmarks.
- In Appendix C, we present additional ablation studies through the lens of PL exponent alpha analysis [44, 46]. These studies further demonstrate how each mechanism of Rep-MTL contributes to facilitating cross-task positive transfer while preserving task-specific learning patterns, thereby mitigating the negative transfer in MTL.
- In Appendix D, we conduct experiments to validate Rep-MTL's robustness and practical applicability, with particular emphasis on the sensitivity of hyper-parameters $\lambda_{tsr}, \lambda_{csa}$, learning rates, and its optimization speed.

A. Implementation Details

This appendix section provides an expansion of the experimental configurations and implementation specifications of the experiments from the main manuscript. We detail the network architectures, optimizers, and training recipes for each included benchmark to ensure reproducibility.

NYUv2 Dataset Following the implementations in previous studies [28, 30], we employ the DeepLabV3+ [5] network architecture, containing a dilated ResNet 50 [17] backbone pre-trained on ImageNet and the Atrous Spatial Pyramid Pooling (ASPP) as task-specific decoders. The MTL model is trained for 200 epochs using the Adam optimizer with an initial learning rate of 10^{-4} and weight decay of 10^{-5}. Consistent with prior works [28, 30], we implement a learning rate schedule where the rate is halved to 5×10^{-5} after 100 training epochs. For the three tasks on NYUv2 [59], we utilize cross-entropy loss for semantic segmentation, L_1 loss for depth estimation, and cosine loss for surface normal prediction. We adopted the same logarithmic transformation as in previous works [30, 34, 49]. During training, all input images are resized to 288×384, and we set the batch size to 8. The experiments are implemented with PyTorch and executed on NVIDIA A100-80G GPUs.

Cityscapes Dataset The implementations for Cityscapes benchmark demonstrate substantial alignment with the one on NYUv2 [28, 30]. Specifically, we adopt the identical DeepLabV3+ [5] architecture, leveraging an ImageNet-pretained dilated ResNet 50 network as the backbone, while the ASPP module serves as task-specific decoders. For model optimization, we establish a 200-epoch training regime utilizing Adam optimizer, with the initial learning rate of 10^{-4} and weight decay of 10^{-5}. The learning rate undergoes a scheduled reduction to 5×10^{-5} upon reaching the 100-epoch milestone. We maintain consistency of loss functions with NYUv2: cross-entropy loss and L_1 loss are employed for semantic segmentation and depth estimation, respectively. We also adopted logarithmic transformation as in previous studies [30, 34, 49]. Throughout the training process, all input images are resized to 128×256, and we utilize a batch size of 64. The experiments are implemented with PyTorch on NVIDIA A100-80G GPUs.

Office-Home Dataset Building upon established protocols from prior works [28, 30], we implement an ImageNet-pretrained ResNet-18 network architecture as the shared backbone, complemented by a linear layer serving as task-specific decoders. In pre-processing, all input images are resized to 224×224. The batch size and the training epoch are set to 64 and 100, respectively. The optimization process employs the Adam optimizer with the learning rate of 10^{-4} and the weight decay of 10^{-5}. We utilize cross-entropy loss for all classification tasks, with classification accuracy serving as the evaluation metric. We also adopted logarithmic transformation as in previous studies [30, 34, 49]. The "Avg." reported in the main manuscript represents the mean performance gains across three independent tasks, which is notably excluded from the calculation of overall task-level performance gains. The experiments are implemented with PyTorch and executed on NVIDIA A100-80G GPUs.

Office-31 Dataset The configurations on Office-31 [52] dataset exhibit notable parallels with the ones on Office-Home [28, 30]. Concretely, we deploy a ResNet-18 network architecture pre-trained on the ImageNet dataset as the shared backbone, complemented by task-specific linear layers for classification outputs. The data processing pipeline standardizes input images to 224×224, while the training protocol extends across 100 epochs with a fixed batch size of 64. The Adam optimizer configured with a learning rate of 10^{-4} and the weight decay of 10^{-5} is used. The cross-entropy loss is used for all the tasks and classification accuracy is used as the evaluation metric. We adopted logarithmic transformation as in previous studies [30, 34, 49]. The "Avg." reported in the main manuscript represents the mean performance gains across three independent tasks, which is notably excluded from the calculation of overall task-level performance gains. The experiments are implemented with

Table 5. Performance on Office-31 dataset with 3 diverse image classification tasks. ↑ indicates the higher the metric values, the better the methods' performance. The best and second-best results of each metric are highlighted in **bold** and underline, respectively.

Method	Amazon	DSLR	Webcam	Avg.↑	Δp_{task}↑
Single-Task Baseline	**86.61**	95.63	96.85	93.03	0.00
EW	83.53	97.27	96.85	$92.55_{\pm 0.62}$	$-0.61_{\pm 0.67}$
GLS [10]	82.84	95.62	96.29	$91.59_{\pm 0.58}$	$-1.63_{\pm 0.61}$
RLW [28]	83.82	96.99	96.85	$92.55_{\pm 0.89}$	$-0.59_{\pm 0.95}$
UW [22]	83.82	97.27	96.67	$92.58_{\pm 0.84}$	$-0.56_{\pm 0.90}$
DWA [35]	83.87	96.99	96.48	$92.45_{\pm 0.56}$	$-0.70_{\pm 0.62}$
IMTL-L [34]	84.04	96.99	96.48	$92.50_{\pm 0.52}$	$-0.63_{\pm 0.58}$
IGBv2 [12]	84.52	98.36	98.05	$93.64_{\pm 0.26}$	$+0.56_{\pm 0.25}$
MGDA [13]	85.47	95.90	97.03	$92.80_{\pm 0.14}$	$-0.27_{\pm 0.15}$
GradNorm [8]	83.58	97.26	96.85	$92.56_{\pm 0.87}$	$-0.59_{\pm 0.94}$
PCGrad [66]	83.59	96.99	96.85	$92.48_{\pm 0.53}$	$-0.68_{\pm 0.57}$
GradDrop [9]	84.33	96.99	96.30	$92.54_{\pm 0.42}$	$-0.59_{\pm 0.46}$
GradVac [63]	83.76	97.27	96.67	$92.57_{\pm 0.73}$	$-0.58_{\pm 0.78}$
IMTL-G [34]	83.41	96.72	96.48	$92.20_{\pm 0.89}$	$-0.97_{\pm 0.95}$
CAGrad [32]	83.65	95.63	96.85	$92.04_{\pm 0.79}$	$-1.14_{\pm 0.85}$
MTAdam [42]	85.52	95.62	96.29	$92.48_{\pm 0.87}$	$-0.60_{\pm 0.93}$
Nash-MTL [49]	85.01	97.54	97.41	$93.32_{\pm 0.82}$	$+0.24_{\pm 0.89}$
MetaBalance [19]	84.21	95.90	97.40	$92.50_{\pm 0.28}$	$-0.63_{\pm 0.30}$
MoCo [15]	84.33	97.54	98.33	93.39	-
Aligned-MTL [55]	83.36	96.45	97.04	$92.28_{\pm 0.46}$	$-0.90_{\pm 0.48}$
IMTL [34]	83.70	96.44	96.29	$92.14_{\pm 0.85}$	$-1.02_{\pm 0.92}$
DB-MTL [30]	85.12	**98.63**	98.51	$94.09_{\pm 0.19}$	$+1.05_{\pm 0.20}$
Rep-MTL (EW)	85.93	98.54	**98.67**	**94.38**$_{\pm 0.53}$	**+1.31**$_{\pm 0.58}$

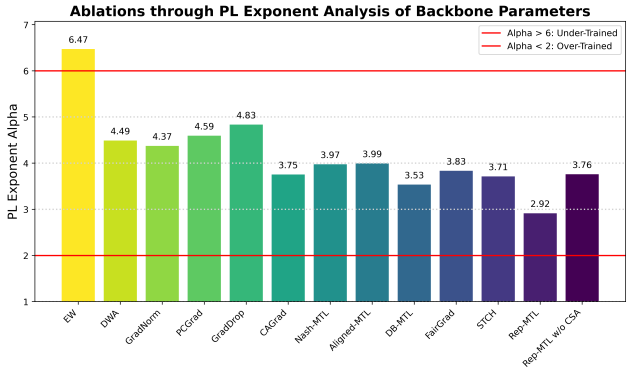

Figure 5. Ablation studies through PL exponent metrics [46] for shared parameters in backbones trained with or without cross-task saliency alignment (notated as "Rep-MTL w/o CA") on NYUv2 [59]. The PL exponent alpha quantifies how well the backbone adapts to the overall MTL objectives, where lower values indicate more effective training. Values outside the range $[2, 6]$ suggest potential over- or under-training due to the insufficient cross-task positive transfer. We leverage this measurement to validate the effectiveness of the cross-task saliency alignment mechanism in our proposed Rep-MTL, as well-trained backbones suggest beneficial information sharing to the overall MTL objectives.

PyTorch and executed on NVIDIA A100-80G GPUs.

B. Office-31 Image Classification Results

This appendix section provides a thorough discussion of our experimental results on Office-31 [52] dataset, presenting detailed observations of performance that were omitted from the main text due to space limitations.

As shown in Table 5, Rep-MTL achieves the highest overall performance among all compared MTO methods. It obtains an average accuracy (**Avg.**↑) of 94.38% and a total performance gain (Δp_{task} ↑) of $+1.31\%$ over the single-task learning (STL) baseline. This result surpasses the next-best method, DB-MTL, which achieves a gain of $+1.05\%$, and stands in stark contrast to the Equal Weighting (EW) baseline that suffers from negative transfer among tasks ($\Delta = -0.61\%$). This demonstrates Rep-MTL's superior ability to effectively manage multi-domain learning on Office-31.

In addition, task-specific performance reveals several notable findings. First, on both the **Webcam** and **Amazon** domains, Rep-MTL achieves competitive accuracies of 98.67% and 85.93%, respectively. Its performance on the challenging **Amazon** domain is particularly noteworthy, outperforming the strong DB-MTL [30] baseline by a significant margin of $+0.81\%$. This improvement is particularly significant due to the varying lighting conditions and image quality. Second, on **DSLR** domain, Rep-MTL delivers a competitive accuracy of 98.54%, narrowly trailing DB-MTL [30] (98.63%) in a tightly contested result.

These results offer key insights into the strengths and limitations of Rep-MTL. On one hand, Rep-MTL demonstrates capabilities to handle multiple tasks effectively, consistently achieving balanced and top-tier performance gains across different tasks. The substantial gains on the **Amazon** and **Webcam** tasks more than compensate for the marginal difference on **DSLR**, leading to the best overall average. On the other hand, however, this balanced approach comes with a trade-off: while Rep-MTL avoids significant performance degradation in task-specific performance compared to existing methods, it may not consistently achieve significant gains across all sub-tasks simultaneously. This observation is particularly evident in the results of the **DSLR** task on Office-31 [52] dataset, where Rep-MTL achieves strong but not leading performance.

Overall, the experimental results suggest that while Rep-MTL has successfully advanced the state-of-the-art in challenging multi-task dense prediction benchmarks, there remains scope for further enhancement. Future research directions could focus on developing mechanisms to maintain the current balanced performance with explicit information sharing while pushing the boundaries of task-specific excellence. This could potentially involve exploring more complex cross-task interactions or adaptive optimization strategies that can better leverage task-specific characteristics.

C. Ablations with PL Exponent Alpha Metrics

While our analysis in Section 4.3 demonstrates Rep-MTL's overall effectiveness in achieving effective multi-task learning—facilitating positive cross-task information sharing

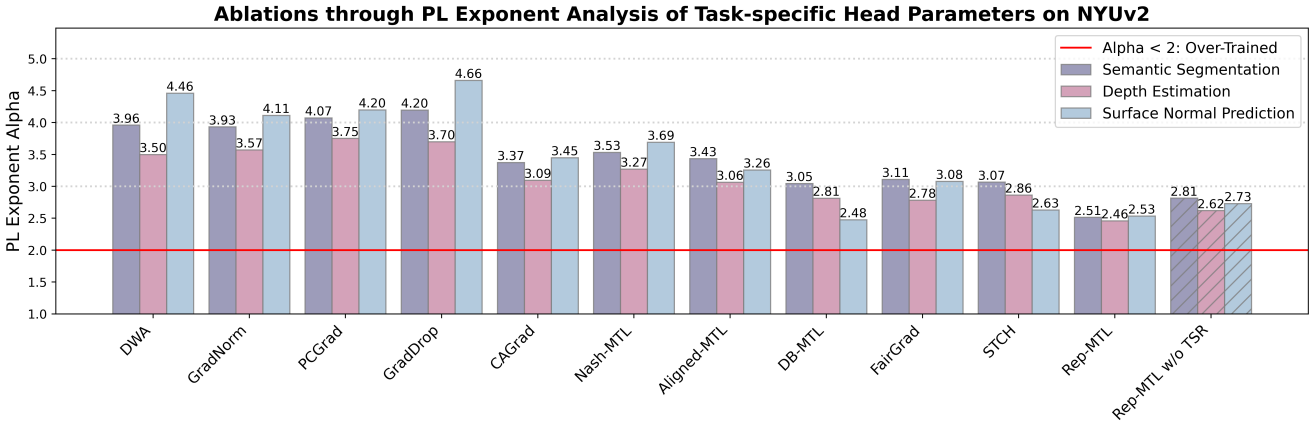

Figure 6. Ablation studies through PL exponent metrics [44, 46] for parameters in diverse decoders trained with or without task-specific saliency regulation in Rep-MTL (notated as "Rep-MTL w/o TR") on NYUv2 [59]. The PL exponent alpha quantifies how well each decoder adapts to its task-specific objective, where lower values indicate more effective training. Values outside the range $[2, 6]$ suggest potential over- or under-training due to task conflicts. The variation across different heads of each method indicates training imbalance. We leverage this measurement to validate the effectiveness of task-specific saliency regulation in Rep-MTL, as well-trained decoders should exhibit both *low and balanced* metric values, indicating successful negative transfer mitigation while preserving task-specific information. The results show that task-specific saliency regulation effectively helps task-specific learning and yields superior and more balanced metrics.

while preserving task-specific patterns for negative transfer mitigation—it does not isolate the contributions of individual components. This appendix section thus presents an additional empirical evaluation of Rep-MTL's two key mechanisms: Cross-Task Saliency Alignment (CA) and Task-specific Saliency Regularization (TR). We first introduce the practical implications of this metric, followed by ablation studies examining each component's effectiveness and distinct contribution to Rep-MTL's overall performance.

C.1. Power Law (PL) Exponent Alpha Analysis

To rigorously evaluate the effectiveness of Rep-MTL's components beyond commonly-used performance metrics, we employ Power Law (PL) exponent alpha [44, 46], a theoretically grounded measure from Heavy-Tailed Self-Regularization (HT-SR) theory [41, 45]. It provides a systematic framework for analyzing the representation capacity and overall learning quality of deep neural networks. In particular, PL exponent alpha is computed for each layer's weight matrix W by fitting the Empirical Spectral Density (ESD) of its correlation matrix $X = W^T W$ to a truncated Power Law distribution: $\rho(\lambda) \sim \lambda^{-\alpha}$, where $\rho(\lambda)$ denotes the ESD, and λ represents eigenvalues of correlation matrix.

Empirical studies have established that well-trained neural networks typically exhibit PL exponent values within the range $\alpha \in [2, 4]$. Values outside this range often indicate suboptimal training dynamics: specifically, $\alpha < 2$ suggests insufficient learning, while $\alpha > 6$ indicates potential over-parameterization or training instabilities. This characteristic makes the PL exponent particularly valuable for assessing training effectiveness across different network architectures and optimization strategies.

In the context of multi-task learning, this metric offers unique insights into both cross-task knowledge transfer and task-specific learning patterns. In particular, for shared backbone parameters, lower alpha values (within the optimal range) typically indicate effective cross-task information sharing, suggesting successful optimization toward the overall MTL objectives. For task-specific heads, balanced and moderately low alpha values across different tasks suggest the preservation of task-specific patterns while minimizing negative transfer effects. Built upon this view, we can systematically evaluate how each component in Rep-MTL contributes to achieving optimal MTL dynamics.

C.2. Effects of Cross-Task Saliency Alignment

Similar to the empirical analysis in Section 4.3, we analyze the effectiveness of Cross-Task Saliency Alignment by examining PL exponent alpha of the DeepLabV3+ backbone parameters on NYUv2 [59] dataset.

As shown in Figure 5, models trained with our cross-task alignment mechanism exhibit alpha values within the optimal range of $[2, 4]$, indicating well-learned and generalizable model parameters in the shared backbone, comparing models trained with and without this alignment mechanism. This demonstrates the effectiveness of our Cross-Task Saliency Alignment for positive information sharing.

C.3. Effects of Task-specific Saliency Regulation

To evaluate the impact of Task-specific Saliency Regulation, we examine the PL exponent alpha of parameters in the DeepLabV3+ task decoder parameters on NYUv2 [59].

As illustrated in Figure 6, the result reveals that models employing our regulation mechanism demonstrate al-

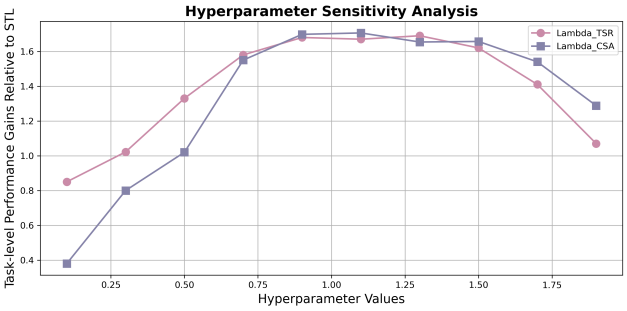

Figure 7. Hyper-parameter sensitivity analysis of our Rep-MTL on NYUv2 [59] dataset. We empirically evaluate the impact of two critical hyper-parameters, λ_{tsr} and λ_{csa}, by fixing one as $\lambda = 0.9$ while varying the other one across a comprehensive range as $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9\}$. The results demonstrate that Rep-MTL maintains stable and competitive performance $\Delta \mathrm{p}_{\text{task}}$ over a substantial range $(0.7, 0.9, 1.1, 1.3, 1.5)$, indicating its robust insensitivity to hyper-parameter variations.

pha values consistently within the optimal range and exhibit more balanced values across all task-specific heads. This balanced distribution suggests the successful preservation of task-specific features while avoiding over-specialization or interference between tasks $(2.60, 2.63, 2.45)$. In contrast, models trained with Rep-MTL without this regulation mechanism exhibit poor and more dispersed PL exponent alpha across decoders $(2.89, 2.74, 2.59)$. This wider variation indicates potential negative transfer and suboptimal task-specific learning. The consistency of alpha values across different task heads in regulated models provides strong evidence that the Task-specific Saliency Regulation in Rep-MTL effectively maintains task-specific patterns.

D. Additional Empirical Analysis

This appendix section presents an empirical investigation designed to further validate the effectiveness and robustness of Rep-MTL. We conduct empirical analyses of hyper-parameter sensitivity and computational efficiency to provide insights into the practical deployment considerations.

D.1. Analysis of Hyper-parameter Sensitivity

We systematically evaluate Rep-MTL's sensitivity to its two primary hyper-parameters, λ_{tsr} and λ_{csa}, on the NYUv2 [59] dataset. Figure 7 illustrates the task-level performance gains relative to STL baselines ($\Delta \mathrm{p}_{\text{task}}$) across various hyper-parameter configurations. Our analysis involves fixing one hyper-parameter at 0.9 while varying the other one across a comprehensive range: $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9\}$. For example, when evaluating the sensitivity of hyper-parameter λ_{tsr}, when fixing the $\lambda_{csa} = 0.9$ then conduct a series of experiments. All experiments are conducted on NVIDIA A100-80G GPUs to ensure consistent evaluation conditions.

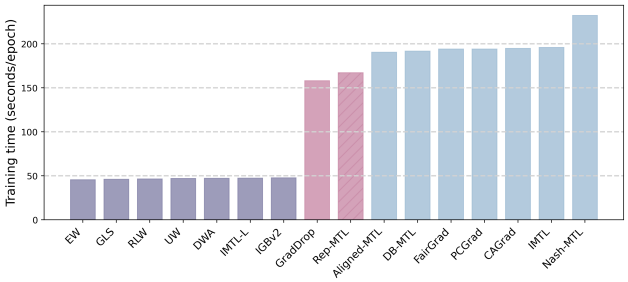

Figure 8. Training time per epoch comparison across different MTL optimization methods on NYUv2 [59]. Methods are categorized into three training efficiency tiers (indicated by different colors), highlighting the inherent trade-off between computational speed and optimization effectiveness in MTL scenarios.

The results reveal several key findings: First, Rep-MTL demonstrates great stability across a wide range of hyper-parameter combinations, particularly within the range of $\{0.7, 0.9, 1.1, 1.3, 1.5\}$ for both λ_{tsr} and λ_{csa}. Second, the method consistently achieves positive performance gains ($\Delta \mathrm{p}_{\text{task}} > 0$) across most hyper-parameter settings, indicating robust improvement over STL baselines. Third, Cross-task Saliency Alignment (CSA) in Rep-MTL, controlled by λ_{csa}, acts as a crucial component. While small values of λ_{csa} lead to suboptimal performance, increasing it beyond a certain threshold demonstrates a significant impact on the overall MTL performance. Based on these observations, we conducted grid search over $\{0.7, 0.9, 1.1, 1.3, 1.5\}$ for both λ_{tsr} and λ_{csa} to determine optimal configurations for all datasets in this paper.

D.2. Analysis of Training Time

To further evaluate the efficiency of Rep-MTL, we conduct a runtime empirical analysis on NYUv2 [59] dataset. Figure 8 presents the average per-epoch training time across different MTL optimization methods, with all experiments conducted over 100 epochs on NVIDIA A100-80G GPUs. Our analysis reveals that Rep-MTL achieves a comparatively favorable balance between training speed and optimization effectiveness. While it requires more training resources than loss scaling methods due to the computation of task saliencies as task-specific gradients in the representation space, it demonstrates superior efficiency compared to most gradient manipulation methods. This increased cost is inherent to approaches requiring second-order (gradient) information, representing a fundamental trade-off and room for further improvement in MTL optimization.

D.3. Analysis of Learning Rate Scaling

Recent studies [64] suggest that different choice of learning rate may impose a strong impact on MTO methods performance. To further demonstate Rep-MTL's robustness, we conduct experiment of learning rate sensitivity on

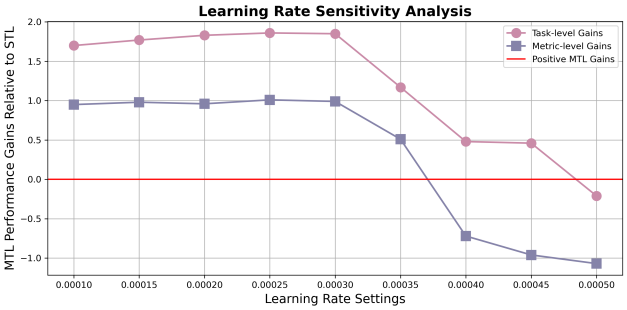

Figure 9. Learning rate sensitivity analysis of our proposed Rep-MTL on NYUv2 [59] dataset. To evaluate the impact of learning rate variations, we systematically scale the learning rate from the default benchmark setting of $1e-4$ to $5e-4$, using a step size of $5e-5$. For each setting, we report the task-level ($\Delta \mathrm{p}_{\text{task}}$) and metric-level ($\Delta \mathrm{p}_{\text{metric}}$) performance gains. Each experiment is repeated three times. The results show that Rep-MTL maintains stable and competitive $\Delta \mathrm{p}_{\text{task}}$ and $\Delta \mathrm{p}_{\text{metric}}$ over a substantial range, indicating its favorable robustness to learning rate variations.

NYUv2 [59] with diverse learning rate settings, as illustrated in Figure 9. Specifically, we scale the learning rate from the default benchmark setting of $1e-4$ to $5e-4$ with a step size of $5e-5$. For each setting, we measure the task-level ($\Delta \mathrm{p}_{\text{task}}$) and metric-level ($\Delta \mathrm{p}_{\text{metric}}$) performance gains. The results show that Rep-MTL maintains stable and competitive $\Delta \mathrm{p}_{\text{task}}$ and $\Delta \mathrm{p}_{\text{metric}}$ over a substantial range, indicating its favorable robustness to learning rate variations.