RUTaL: Residual Upcycling with Task Ladder for Efficient Multi-Task Learning

Haiming Yao[1,2], Wei Luo[1,2], Qiyu Chen[2], Jianxing Liao[2], and Wei You[2⋆]

[1] Department of Precision Instruments, Tsinghua University, China
[2] Advanced Computing and Storage Lab, Huawei Technologies Co. Ltd, China
{yhm22, luow23}@mails.tsinghua.edu.cn
{chenqiyu13, liaojianxing, youwei22}@huawei.com

Abstract. Pretrained vision models are widely used as backbones for downstream tasks, yet adapting them to multi-task learning (MTL) remains challenging. Naive full fine-tuning incurs substantial overhead and often exacerbates task interference, while existing parameter-efficient fine-tuning (PEFT) methods are largely designed for single-task adaptation. We propose Residual Upcycling with Task Ladder (RUTaL), a parameter-efficient multi-task vision adaptation framework. To accommodate the diverse and heterogeneous representation demands in MTL, We proposed Task-General Residual Upcycling (TGRU) to transform a pre-trained vision Transformer backbone into a Mixture-of-Experts architecture through low-rank residual weight reparameterization, enabling efficient and task-scalable capacity expansion. Built upon this upcycled representation paradigm, we further introduce Task-Specific Ladder Adaptation (TSLA), which extracts task-relevant features from the shared upcycled representations in a decoupled manner to accommodate the unique requirements of each task. Experiments on multi-task dense scene understanding benchmarks show that RUTaL achieves state-of-the-art performance while demonstrating superior computational efficiency.

Keywords: Multi-Task Learning · Parameter-Efficient Fine-Tuning· Decoupled Model Adaptation

1 Introduction

The modern era pursues the development of vision models that learn general-purpose visual representations through large-scale pretraining [41,50]. By acquiring rich and transferable features, these backbone models have become powerful initializations for a wide range of downstream vision tasks [17, 44]. Traditional adaptation methods fully fine-tune backbone models end-to-end by attaching task-specific decoders for target vision tasks, such as semantic segmentation [32], object detection [8], and depth estimation [2]. However, training and storing these high-capacity and heavily structured backbone models demands substantial computational resources, particularly in scenarios involving multiple downstream tasks.

⋆ Corresponding author.

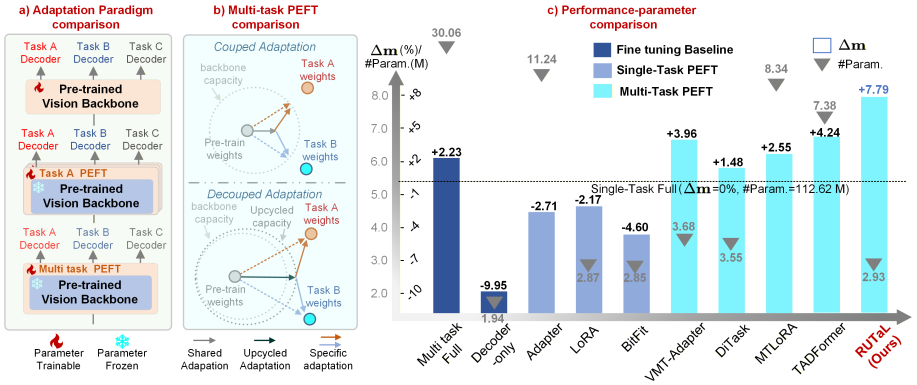

Fig. 1: (a) Comparison of different adaptation paradigms in multi-task scenarios. (b) Illustration of the motivations behind existing coupled PEFT methods for MTL and our proposed decoupled paradigm. (c) Comparison of overall performance ($\Delta\mathbf{m}$) and trainable parameters of different adaptation methods on the PASCAL-Context dataset.

To enable the effective use of pre-trained models under constrained computational budgets, resource-efficient adaptation methods—particularly Parameter-Efficient Fine-Tuning (PEFT) [72]-have recently emerged, achieving task-specific customization by updating only a small subset of parameters. Representative families of such approaches include the low-rank adaptation series [23], adapter-based series [52], and prompt tuning series [37], which were first popularized in large language models (LLM) [10] and have been extended to the vision [7,16,82].

In scenarios involving multiple correlated downstream tasks, multi-task learning (MTL) [80] can improve model efficiency by enabling multiple tasks to share a common backbone representation. However, the presence of multiple learning objectives in MTL significantly increases training complexity and resource demands for full fine-tuning methods (Fig. 1(a, top)), making PEFT particularly appealing for reducing adaptation overhead. Despite this advantage, most early PEFT methods [15, 19, 23] are primarily designed for single-task adaptation. When naively extended to more complex multi-task settings, the straightforward per-task adaptation strategy (Fig. 1(a, middle)) often becomes sub-optimal, as they do not explicitly model key characteristics of MTL, such as cross-task knowledge sharing and the mitigation of negative transfer, thereby leading to an inherent mismatch between existing PEFT paradigms and multi-task objectives.

As shown at the bottom of Fig. 1(a), recent efforts to extend PEFT to MTL [1, 3, 40, 62] have explored various mechanisms for modeling cross-task correlations and task-aware dynamics. Polyhistor [40] employs hyper-networks to generate task-specific adapter parameters. VMT-Adapter [62] enables cross-task interaction through shared projection layers while extracting task-specific knowledge. MTLoRA [1] introduces both task-agnostic and task-specific low-rank adaptations, and TAD-Former [3] models input-conditioned dynamic semantics. Despite these advances, we observe that existing methods follow a cou-

pled learning paradigm, as shown at the top of Fig. 1(b), where shared and task-specific adaptations (solid arrows) are entangled within a fixed-capacity backbone. Consequently, both adaptations are optimized in the same representational space, competing for limited backbone capacity, while task adaptation (dashed arrows) remains constrained by the backbone capacity (dashed circle).

In contrast, we advocate a decoupled adaptation paradigm for MTL as shown at the bottom of Fig. 1(b). Rather than co-optimizing shared and task-specific updates within a single representational space, we argue that effective multi-task adaptation should be grounded in a capacity-aware principle. Specifically, task-generic and task-specific adaptations should be organized into distinct representational pathways. We first upcycle the backbone in a task-generic manner (dark gray dashed circle) to establish a shared foundation applicable to multiple tasks, forming a structured multi-task representation (solid black arrow). Subsequently, task-specific adaptive representations are derived from this shared foundation through a decoupled space, enabling flexible task specialization while ensuring non-interference between the shared and task-specific representations.

According to this philosophy, we propose Residual Upcycling with Task Ladder (RUTaL), a novel MTL-oriented PEFT approach. To achieve decoupling adaptation, we first introduce Task-General Residual Upcycling (TGRU), which transforms a pre-trained Transformer backbone into a task-generic Mixture-of-Experts (MoE) architecture via low-rank residual weight reparameterization. Rather than fully re-optimizing the pre-trained parameters, this upcycling strategy incrementally expands the shared backbone representational space, enabling capacity scaling while preserving parameter efficiency. Built upon this upcycled backbone, we further design Task-Specific Ladder Adaptation (TSLA), a lightweight decoupled side branch that selectively extracts and refines task-relevant representations from the shared space. By isolating task specialization from general backbone capacity expansion, RUTaL alleviates representational competition and enables flexible multi-task adaptation without entangling shared and task-specific updates. Our core contributions are as follows:

- We propose RUTaL, a novel decoupled PEFT framework for MTL that separates task-generic backbone capacity expansion from a task-specific adaptation pathway, mitigating representational competition and enabling more structured multi-task adaptation.
- We introduce Task-General Residual Upcycling (TGRU), which converts a pre-trained Transformer backbone into a task-generic MoE via low-rank residual weights, enabling capacity expansion with high parameter efficiency.
- We design Task-Specific Ladder Adaptation (TSLA), a lightweight decoupled side pathway that selectively extracts task-relevant representations without interfering with the shared backbone.
- We evaluate RUTaL on two widely used MTL benchmarks, PASCAL-Context and NYUD v2, where it achieves superior performance with significantly fewer trainable parameters (Fig. 1(c)) and improved computational efficiency.

2 Related Work

2.1 Multi-Task Learning

Multi-task learning (MTL) [73, 80] improves parameter and computational efficiency by training a single model to perform multiple tasks, focusing on learning shared representations while maintaining task-specific discrimination [67, 68]. Common research directions include encoder–decoder architecture design [4, 24, 27, 65, 78], optimization-oriented methods [1, 39, 74], and multi-task model fusion [57, 58, 66]. Among them, architectural design methods are the most widely studied, in which Encoder-centric methods [42, 48, 53] emphasize feature sharing for efficiency, while decoder-centric methods [59, 67, 69, 81] explicitly model task differences to reduce interference. More recently, Vision Foundation Models (VFMs) have been integrated into MTL frameworks [43, 79], leveraging their general-purpose visual representations that generalize effectively across diverse tasks. However, most architecture-centric approaches are built upon pre-trained backbones and rely on full fine-tuning for downstream adaptation. Modern pretrained backbones—particularly VFMs—are typically built with a heavy structure, making such adaptation computationally and memory intensive.

2.2 Parameter Efficient Fine-Tuning

PEFT [20] emerged in LLMs to mitigate the prohibitive cost of full fine-tuning while preserving transfer performance. Early approaches include adapter-based tuning [21, 52], bias-only updating [75], prompt-based methods [34, 37], and low-rank reparameterization exemplified by LoRA [23]. With the rise of vision transformers (ViTs) [11, 41], PEFT has been extended to vision through approaches such as AdaptFormer [5], Visual Prompt Tuning [26], and other parameter-efficient adaptation strategies for ViTs [63].

Recent studies extend PEFT to MTL [31, 46]. Polyhistor [40] employs hypernetworks to generate task-conditioned adapters. VMT-Adapter [62] promotes structured cross-task interaction through shared adapter components. MTLoRA [1] decomposes adaptation into task-agnostic and task-specific low-rank components, while TADFormer [3] introduces task-adaptive dynamic transformers. Although these methods improve parameter efficiency in MTL, they generally adopt a coupled adaptation strategy in which shared and task-specific components are learned within a fixed-capacity pre-trained backbone. Such a design can induce representational competition and limit scalability across tasks.

2.3 Model Upgrading and Branching

Recent works investigate structural strategies for enhancing pretrained models, primarily along two directions: upgrading backbone capacity to strengthen shared representations and introducing branch-based adaptation modules to capture task-specialized patterns.

Among upgrading approaches, model upcycling [13, 25, 33] has emerged as a principled strategy for scalable capacity expansion. It transforms dense pretrained networks into Mixture-of-Experts (MoE) architectures while preserving pretrained knowledge. Sparse upcycling [33] demonstrates that dense checkpoints can be restructured into MoE-based models without training from scratch. Subsequent works extend this paradigm to language and vision models [18, 35, 83, 84], showing that pretrained Transformers can be systematically converted into sparse MoE variants with improved scalability and computational efficiency.

Complementary to backbone upgrading, branch-based adaptation enables task-specific modeling through decoupled auxiliary pathways, thereby reducing interference in the shared backbone. Side-Tuning [77] formulates adaptation as learning additive side branches parallel to a frozen backbone, thereby decoupling specialization from shared representations. Later works extend this paradigm with structured side adapters [6, 64], ladder-style tuning mechanisms [56], convolutional bypass modules for vision transformers [28], and universal parallel tuning frameworks [9]. Recent studies further demonstrate that branch-based designs can simultaneously improve parameter, memory, and time efficiency across diverse adaptation scenarios [14, 47, 71].

3 RUTaL Methodology

3.1 Overview

The overall architecture of the proposed RUTaL framework for efficient MTL is illustrated in Fig. 2. RUTaL adopts a structurally decoupled design. To accommodate the growing number and heterogeneity of tasks in MTL, we perform Task-General Residual Upcycling (TGRU) on a pretrained ViT backbone to expand its representational capacity. Specifically, the backbone is transformed into a MoE–based shared branch, which upgrades model capacity and obtains strong task-general representations. To further capture task-specific characteristics that cannot be fully modeled by the shared branch, we introduce the Task-Specific Ladder Adaptation (TSLA), which designed to extract specialized features for each task. These task-adaptive features are subsequently fed into their corresponding decoders for final prediction. In the following sections, we present the detailed design and underlying mechanisms of each component.

3.2 Task-General Residual Upcycling

Scaling laws [29] in LLMs indicate that increasing model capacity is critical for learning general representations that are transferable across downstream tasks. This insight naturally motivates expanding model capacity to better capture cross-task shared representations, thereby improving MTL performance. To this end, we propose Task-General Residual Upcycling (TGRU), an efficient capacity expansion strategy that upgrades a pretrained Transformer backbone into an task-general MoE architecture.

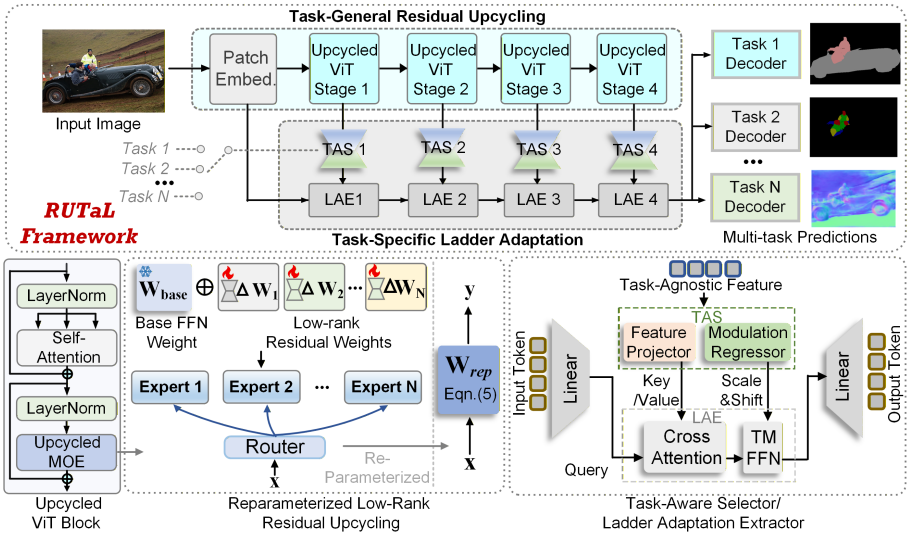

Fig. 2: Schematic of the proposed RUTaL. The framework consists of two key components: Task-General Residual Upcycling (TGRU), which converts a pretrained ViT into a task-agnostic backbone via reparameterized low-rank residual experts, and Task-Specific Ladder Adaptation (TSLA), which progressively extracts and refines task-specific features from the shared representations in a stage-wise manner.

Preliminary Upcycled Mixture-of-Experts. Model upcycling [33] provides a parameter-efficient pathway to scale pretrained Transformers into MoE architectures without retraining from scratch. Consider a pretrained ViT composed of multiple blocks, each consisting of multi-head attention (MHA), layer normalization (LN), and a feed-forward network (FFN). In model upcycling, the weights of MHA and LN are directly inherited without modification. In contrast, the FFN weights $\mathbf{W}_{\text{base}} = \{W^{\text{up}}_{\text{base}} \in \mathbb{R}^{D\times H}, W^{\text{down}}_{\text{base}} \in \mathbb{R}^{H\times D}\}$, where D and H denote the input and hidden dimensions, respectively, are replicated N times to initialize N expert networks $\{E_1, E_2, \ldots, E_N\}$. Subsequently, a router network $R \in \mathbb{R}^{D\times N}$ is randomly initialized to form a standard MoE layer. Given an input token $\mathbf{x}$, the output $\mathbf{y}$ is expressed as:

$$\mathbf{y} = \sum_{i=1}^{N} \text{Softmax}(R(\mathbf{x}))_i \times E_i(\mathbf{x}), \tag{1}$$

where $E_i(\mathbf{x}) = W_i^{\text{down}}\sigma(W_i^{\text{up}}\mathbf{x})$, with σ denoting the activation function. The Model Upcycling is typically performed via full-parameter training. After training, the weights of the N experts are updated to $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_N\}$.

Efficient Low-rank Residual Upcycling. After training, a MoE layer obtained from conventional upcycling needs to store the updated weights of all N experts, which substantially compromises parameter efficiency. As shown in in the lower-left panel of Fig 2, to alleviate this limitation, we propose an efficient

low-rank residual strategy. Specifically, since the expert weights $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_N\}$ are all optimized from the same initial parameters $\mathbf{W}_{\text{base}}$, each expert weight can be expressed as:

$$\mathbf{W}_i = \mathbf{W}_{base} + \Delta\mathbf{W}_i \tag{2}$$

This implies that $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_N\}$ can be decomposed into a shared base weight $\mathbf{W}_{\text{base}}$ and N expert-specific residual weights $\{\Delta\mathbf{W}_1, \Delta\mathbf{W}_2, \ldots, \Delta\mathbf{W}_N\}$. Therefore, the expert weights obtained via full-parameter training in preliminary model upcycling can be equivalently represented by lightweight residual weights. Inspired by low-rank adaptation (LoRA) [23], we parameterize each residual weight using a pair of low-rank matrices as

$$\Delta\mathbf{W}_i = \left\{\Delta W_i^{\text{up}} = A_i^{\text{up}} B_i^{\text{up}}, \Delta W_i^{\text{down}} = A_i^{\text{down}} B_i^{\text{down}}\right\}, \tag{3}$$

where $A_i^{\text{up}} \in \mathbb{R}^{D\times r}$, $B_i^{\text{up}} \in \mathbb{R}^{r\times H}$, $A_i^{\text{down}} \in \mathbb{R}^{H\times r}$, and $B_i^{\text{down}} \in \mathbb{R}^{r\times D}$. $r \ll min(D, H)$ indicates the rank. By introducing low-rank residual matrices, the number of parameters required to upcycle the model into N experts is reduced from $N\cdot 2\cdot H\cdot D$ to $N\cdot 2\cdot r\cdot (D+H)$, significantly improving parameter efficiency. The upcycled MoE layer can be formulated as follows:

$$\mathbf{y} = \sum_{n=1}^{N} \text{Softmax}\Big(R(\mathbf{x})_n\Big) \times \Big(W_{base}^{\text{down}} + \Delta W_i^{\text{down}}\Big)\sigma\Big(\left(W_{base}^{\text{up}} + \Delta W_i^{\text{up}}\right)\mathbf{x}\Big). \tag{4}$$

It can be observed that the N experts require N independent forward passes, limiting knowledge sharing across experts. To address this issue, we replace the ΔW_i^{up} with a shared up-projection $\Delta W_{shared}^{\text{up}}$, enabling cross-expert knowledge transfer. Under this design, the MoE layer admits a functional reparameterization with two aggregated matrices, $\mathbf{W}_{rep} = \left\{W_{rep}^{\text{up}}, W_{rep}^{\text{down}}\right\}$:

$$\begin{cases} W_{rep}^{\text{up}} = W_{base}^{\text{up}} + \Delta W_{shared}^{\text{up}}, \\ W_{rep}^{\text{down}} = \displaystyle\sum_{n=1}^{N} \text{Softmax}(R(\mathbf{x})_n)\left(W_{base}^{\text{down}} + \Delta W_i^{\text{down}}\right). \end{cases} \tag{5}$$

From this functional reparameterization perspective, the overall computation can be expressed as $\mathbf{y} = W_{rep}^{\text{down}}\sigma\Big(W_{rep}^{\text{up}}\mathbf{x}\Big)$. Here, W_{rep}^{up} promotes cross-expert knowledge sharing, while W_{rep}^{down} preserves expert specialization through dynamic aggregation. Moreover, the number of parameters is reduced to $(N+1)\cdot r\cdot (D+H)$. Applying the proposed upcycling strategy to all FFN layers yields an upcycled Transformer backbone that produces the task-agnostic representations.

3.3 Task-Specific Ladder Adaptation

Following backbone upcycling, we extract task-specific features from shared representations for downstream tasks $T = t_1, t_2, \ldots, t_K$. Inspired by the conditional modeling paradigm widely adopted in generative [51] and multimodal [36] models, we formulate multi-task learning as task-conditioned adaptation over shared

representations. As shown in Fig. 2, we introduce the Task-Specific Ladder Adaptation (TSLA) branch to query task-relevant information from the shared backbone. Consequently, TSLA enables task specialization without modifying the shared backbone, reducing task interference.

Given the upcycled Transformer backbone, we partition it into multiple stages. As shown in the lower-right panel of Fig. 2, for each stage i, the task-agnostic feature $\mathcal{F}_i$ is first fed into a Task-Aware Selector (TAS), which is specialized for task t and extracts task-relevant information from the shared representation. The selected features are then integrated into the corresponding Ladder Adaptation Extractor (LAE) for interaction.

Task-Aware Selector. The TAS aims to extract task-relevant information from the task-agnostic feature $\mathcal{F}_i$. For each task $t \in T$, a dedicated TAS module is instantiated to produce both task-adapted representations and task-conditioned modulation parameters. Specifically, TAS comprises two lightweight components. The first is a Task Feature Projector $P^t(\cdot)$, which maps $\mathcal{F}_i \in \mathbb{R}^{L\times D}$ to task-adapted key–value features $\mathcal{Z}_i^t = P^t(\mathcal{F}_i) \in \mathbb{R}^{L\times \hat{D}}$, where L denotes the token sequence length, D is the Transformer feature dimension, and $\hat{D}$ represents the feature dimension after projection. The second is a Task Modulation Generator $R^t(\cdot)$, which predicts task-conditional modulation parameters $\{\gamma_i^t \in \mathbb{R}^{L\times \hat{D}}, \beta_i^t \in \mathbb{R}^{L\times \hat{D}}\} = R^t(\mathcal{F}_i)$. For parameter efficiency and simplicity, both $P^t(\cdot)$ and $R^t(\cdot)$ are implemented using a lightweight MLP.

Ladder Adaptation Extractor. After obtaining task-selective information from the TAS, we introduce the Ladder Adaptation Extractor (LAE) to generate task-adaptive features. The LAE module is shared across tasks to maintain parameter efficiency, while task specificity is injected through the task-conditioned inputs provided by the TAS.

Specifically, for each task t, the LAE takes as input the task-adapted feature $\mathcal{A}_{i-1}^t \in \mathbb{R}^{L\times D}$ from the previous LAE (with $\mathcal{A}_0^t$ initialized from the patch embedding), together with the task-conditioned signals $\mathcal{Z}_i^t$ and $\{\gamma_i^t, \beta_i^t\}$ produced by the corresponding TAS. The feature $\mathcal{A}_{i-1}^t$ is first projected through a linear layer to obtain $\hat{\mathcal{A}}_{i-1}^t \in \mathbb{R}^{L\times \hat{D}}$. Subsequently, a cross-attention operation is applied to inject the task-adapted key–value features $\mathcal{Z}_i^t$ into $\hat{\mathcal{A}}_{i-1}^t$:

$$\mathcal{H}_{i-1}^t = \hat{\mathcal{A}}_{i-1}^t + \mathrm{Attention}\big(\mathrm{norm}(\hat{\mathcal{A}}_{i-1}^t), \mathrm{norm}(\mathcal{Z}_i^t)\big), \tag{6}$$

where $\mathrm{norm}(\cdot)$ denotes LayerNorm and $\mathrm{Attention}(\cdot)$ is implemented as sparse attention [7] for computational efficiency. $\mathcal{H}_{i-1}^t$ denotes the resulting feature after cross-attention fusion. We then apply a task-modulated FFN conditioned on $\{\gamma_i^t, \beta_i^t\}$ to further enhance task-specific feature transformation:

$$\hat{\mathcal{A}}_i^t = W^{\mathrm{down}}\Big(\gamma_i^t \odot \sigma\big(W^{\mathrm{up}}\mathrm{norm}(\mathcal{H}_{i-1}^t)\big) + \beta_i^t\Big), \tag{7}$$

where $\odot$ denotes the Hadamard product, and W^{up} and W^{down} are learnable weights. Finally, a linear projection restores $\hat{\mathcal{A}}_i^t$ to the original dimension, yielding $\mathcal{A}_i^t$, which is passed to the next stage.

3.4 Configuration Details and Efficiency Analysis.

Architectural Configurations. Our framework is compatible with existing ViT architectures. TGRU acts as a generic upcycling strategy and is applied to all FFN layers in a pretrained ViT backbone. TSLA requires multi-stage features. For vanilla ViT, we evenly divide the blocks into four stages to extract hierarchical representations. For hierarchical transformers (*e.g.*, Swin Transformer [41]), we directly use their native stage-wise features. In this case, downsampling is introduced in the TSLA branch to ensure stage alignment. For simplicity, we reuse the pretrained downsampling layers and keep them frozen during training. Given the hierarchical task-adaptive features $\{\mathcal{A}_i^t\}_{i=1}^4$, we employ task-specific decoders for each dense prediction task. Following prior work [1, 62], we adopt a simple MLP-based SegFormer [61] decoder for all tasks.

Model Training To train the multi-task model, we optimize a weighted sum of task-specific losses $\mathcal{L} = \sum_{t \in T} \omega_t \mathcal{L}_t$, where ω_t and $\mathcal{L}_t$ denote the weight and loss of task t, respectively. Specifically, we adopt cross-entropy loss for semantic segmentation and human part segmentation, $L1$ loss for surface normal estimation, and balanced cross-entropy loss for saliency detection. The task weights w_i follow the standard MTL configuration for fair comparison [1, 43, 67].

Parameter and Computational Efficiency. Our model is highly efficient in both parameter usage and computation. Within TGRU and TSLA, only the TAS modules are task-specific and scale linearly with the number of tasks, while all other components are shared across tasks. Since each TAS is implemented using lightweight MLPs, the overall framework remains parameter-efficient. When instantiated with Swin Transformer-Tiny [41] as the backbone, the model introduces only 2.93M trainable parameters. Furthermore, TGRU is task-agnostic. When applied to a pretrained Transformer, it enables the backbone to simultaneously support multiple downstream tasks through a single forward pass, resulting in efficient inference.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate our method on two multi-task dense prediction benchmarks, PASCAL-Context [49] and NYUD v2 [55], which include semantic segmentation, human part segmentation, surface normal estimation, saliency detection, and depth estimation. Standard metrics are adopted for evaluation: mIoU for segmentation and saliency tasks, and RMSE for surface normal and depth estimation. We further report $\boldsymbol{\Delta m}$ [45] to quantify the overall multi-task improvement over the baseline. For consistency, single-task full fine-tuning of the Swin-Tiny backbone is used as the baseline. Following the data augmentation protocol of [43], training images are randomly scaled, cropped, and augmented with random horizontal flipping and color jittering. During evaluation, images are resized and padded while preserving the aspect ratio, using the same normalization as in training.

Compared Baselines. We compare our method with a range of baselines, grouped into three categories: (1) *Traditional fine-tuning*: single-task full fine-tuning, where each task uses an independent encoder and decoder with all parameters updated; multi-task full fine-tuning, with a shared encoder and task-specific decoders, updating all parameters; and decoder-only fine-tuning, where only task-specific decoders are trained. (2) *Single-task PEFT methods*, designed for efficient adaptation to individual tasks, including Adapter [19], VL-Adapter [70], LoRA [23], VPT [26], BitFit [76], and Compactor/Compactor++ [30]. (3) *Multi-task PEFT methods*, recently developed for multi-task adaptation, including HyperFormer [31], Polyhistor [40], VMT-Adapter [62], MTLoRA [1], TADFormer [3], and DiTask [46].

Implementation details. Our RUTaL framework is trained using the AdamW optimizer for 300 epochs, with a learning rate of 4×10^{-4}, a weight decay of 1×10^{-4}, and a cosine learning rate schedule. By default, we use a Swin Transformer-Tiny backbone pretrained on ImageNet-1K [41]. The number of experts N is set to 4, the low-rank dimension r is 8, and the projected feature dimension is $\hat{D} = D/4$. Unless otherwise specified, all baselines use the same four-stage backbone features and SegFormer [61] All-MLP decoder for fair comparison; MTLoRA, TADFormer, and DiTask retain their original HRNet [60]-based decoders. Our SegFormer decoder uses an embedding dimension matching the backbone feature dimension, and its prediction head consists of one transposed convolution followed by two convolution layers, with the hidden dimension set to half the embedding dimension. In TSLB, the task feature projector and task modulation generator are implemented as two-layer MLPs with GELU activations, each with a hidden dimension equal to one-fourth of its output dimension. LAE adopts deformable attention [85] with 6 heads, 4 sampling points, and an FFN ratio of 1.

4.2 Main Results

Table 1 reports the results on PASCAL-Context with Swin-Tiny as the backbone. Traditional multi-task full fine-tuning slightly improves $\boldsymbol{\Delta m}$ over single-task training, while decoder-only tuning degrades due to limited adaptability. Single-task PEFT methods generally fail in multi-task settings, often yielding negative $\boldsymbol{\Delta m}$, indicating insufficient representation sharing. In contrast, multi-task PEFT methods consistently outperform these baselines. RUTaL achieves the best overall performance with $\boldsymbol{\Delta m} = +7.79\%$ using only 2.93M trainable parameters, surpassing the strongest baseline TADFormer by $+3.55\%$ while reducing parameters by 4.45M. It attains top or near-top performance on semantic segmentation, human part segmentation, and surface normal estimation, with competitive results on saliency estimation.

Table 2 presents results on NYUDv2. RUTaL consistently outperforms MTLoRA and TADFormer across all tasks, significantly reducing the RMSE of surface normals (21.75 vs. 27.16) and depth (0.6175 vs. 0.6577), while improving semantic segmentation to 42.76 mIoU (vs. 37.30). These gains are achieved with only 2.58M parameters, less than half of TADFormer.

Table 1: Quantitative comparison on four dense prediction tasks on the PASCAL-Context dataset. We compare conventional fine-tuning, single-task PEFT, and multi-task PEFT methods. Our method achieves the best overall performance ($\Delta\mathbf{m}$) with the fewest trainable parameters (#**TP**). **Bold** and underlined entries indicate the best and second-best results, respectively.

Method	Taxonomy	SemSeg (mIoU ↑)	HumPa (mIoU ↑)	Saliency (mIoU ↑)	Normals (RMSE ↓)	$\Delta\mathbf{m}$(%) ↑	#**TP**(M)↓
Traditional Fine-Tuning Methods							
Single-task full fine-tuning	–	67.21	**61.93**	62.35	17.97	0	112.62
Multi-task full fine-tuning	–	67.56	60.24	65.21	16.64	+2.23	30.06
Decoder-only fine-tuning	–	65.09	53.48	57.46	20.69	-9.95	1.94
Single-task PEFT-based Methods							
Adapter [19]	*ICLR 22*	69.21	57.38	61.28	18.83	-2.71	11.24
VL-Adapter [70]	*CVPR 22*	70.21	59.15	62.29	19.26	-1.83	4.74
LoRA [22]	*ICLR 22*	70.12	57.73	61.90	18.96	-2.17	2.87
VPT-shallow [26]	*ECCV 22*	62.96	52.27	58.31	20.90	-11.18	2.57
VPT-deep [26]	*ECCV 22*	64.35	52.54	58.15	21.07	-10.85	3.43
BitFit [76]	*ACL 22*	68.57	55.99	60.64	19.42	-4.60	2.85
Compactor [30]	*NIPS 21*	68.08	56.41	60.08	19.22	-4.55	2.78
Compactor++ [30]	*NIPS 21*	67.26	55.69	59.47	19.54	-5.84	2.66
Multi-task PEFT-based Methods							
HyperFormer [31]	*ACL 21*	71.43	60.73	65.54	17.77	+2.64	75.32
Polyhistor [40]	*NIPS 22*	70.87	59.54	65.47	17.47	+2.34	8.96
VMT-Adapter [62]	*AAAI 24*	<u>71.60</u>	60.67	64.02	<u>16.41</u>	+3.96	3.68
MTLoRA [1]	*CVPR 24*	67.90	59.84	<u>65.40</u>	16.60	+2.55	8.34
DiTask [46]	*CVPR 25*	70.09	59.03	64.55	17.47	+1.48	<u>3.55</u>
TADFormer [3]	*CVPR 25*	70.82	60.45	**65.88**	16.48	<u>+4.24</u>	7.38
RUTaL(Ours)	*ECCV 26*	**73.42**	<u>61.70</u>	64.74	**14.65**	**+7.79**	**2.93**

Table 3 further provides a comprehensive efficiency evaluation on PASCAL-Context from a adaptation perspective. Beyond trainable parameters, we additionally measure GPU memory consumption and training time per batch to assess practical computational cost. RUTaL requires the least training memory (12.67 GB) and achieves the shortest training batch time (0.248 s/batch), reducing memory by 7.80/5.55 GB and batch time by 0.210/0.057 s compared to TADFormer and MTLoRA, respectively. At the same time, it delivers the highest overall performance gain ($\Delta\mathbf{m}$ of +7.79%). These results demonstrate that RUTaL consistently enhances both accuracy and training efficiency, yielding the most favorable overall accuracy–efficiency trade-off among compared methods.

4.3 In-depth analysis

We conduct extensive experiments to investigate the underlying mechanisms of the proposed RUTaL.Unless otherwise specified, all experimental analyses are conducted using the Swin-Tiny backbone on the PASCAL-Context dataset.

Ablation Study. We evaluate the individual contributions of TGRU and TSLA in Table 4. Removing TSLA (Row 1), which directly adapts shared task-

Table 2: Comparison results on the three tasks on the NYUDv2 dataset.

Method	Normals (RMSE ↓)	Depth (RMSE ↓)	SemSeg (mIoU ↑)	#TP (M↓)
MTLoRA	27.38	0.6791	36.33	7.81
TADFormer	<u>27.16</u>	<u>0.6577</u>	<u>37.30</u>	6.47
RUTaL(Ours)	**21.75**	**0.6175**	**42.76**	**2.58**

Table 3: Comprehensive efficiency and performance comparison results.

Method	#TP (M↓)	Memory (GB↓)	Time (s/batch↓)	Δm (% ↑)
MTLoRA	8.34	<u>18.22</u>	<u>0.305</u>	+2.55
TADFormer	<u>7.38</u>	20.47	0.458	<u>+4.24</u>
RUTaL(Ours)	**2.93**	**12.67**	**0.248**	**+7.79**

Table 4: Ablation of the proposed modules.

TGRU	TSLA	#TP	Δm
✓	×	1.76M	+2.42%
×	✓	2.02M	+6.75%
✓	✓	2.93M	+7.79%

Table 5: Ablation of the upcycling strategy in TGRU.

Upcycling	Experts	#TP	Δm
$\mathbf{W}_i$	$N \times 2$	71.13M	+4.72%
$\Delta\mathbf{W}_i$	$N \times 2$	3.45M	+7.60%
$\mathbf{W}_{rep}$	$N + 1$	2.93M	+7.79%

Table 6: Ablation of task adaptation in TSLA.

TAS	LAE	#TP	Δm
$P^t(\cdot)$	CA	2.52M	+6.71%
$R^t(\cdot)$	TM	2.55M	+3.01%
$P^t(\cdot)$&$R^t(\cdot)$	CA&TM	2.93M	+7.79%

agnostic representations from the upcycled backbone, degrades performance, indicating the necessity of explicit task-specific adaptation. Removing TGRU (Row 2), which disables task-agnostic backbone capacity expansion, also leads to suboptimal results, highlighting the importance of enhancing shared representation capacity. Combining TGRU and TSLA yields the best overall performance, validating their complementary roles.

Table 5 further compares different upcycling strategies in TGRU, including fully fine-tuned MoE ($\mathbf{W}_i$), per-expert low-rank residual upcycling ($\Delta\mathbf{W}_i$), and our re-parameterized aggregation upcycling ($\mathbf{W}_{rep}$). Despite introducing substantially more parameters, the fully fine-tuned MoE fails to deliver superior performance. We attribute this to the mismatch between a lightweight router and over-parameterized experts under the limited training data volume, which results in coarse routing boundaries, expert under-utilization, and functional redundancy [12, 38, 54]. Consequently, the increased parameter count does not translate into effective capacity gains. In contrast, the lightweight low-rank residual strategy yields more stable improvements by constraining expert updates to global residual directions applied to the shared base weights, thereby regularizing expert specialization. Furthermore, our re-parameterized aggregation effectively balances shared knowledge and expert-specific functionality, achieving the best performance with the fewest additional expert parameters.

Table 6 analyzes the task adaptation mechanisms within TSLA. TSLA performs task specialization via the Task-Aware Selector (TAS) and extraction through the Ladder Adaptation Extractor (LAE). We evaluate (1) a Task Feature Projector $P^t(\cdot)$ with cross-attention (CS), and (2) Task Modulation (TM) driven by the Task Modulation Generator $R^t(\cdot)$. The results show that integrating both mechanisms yields the best performance while introducing only marginal additional parameters.

Hyperparameter Trade-offs Table 7(a)-(c) investigates the impact of key hyperparameters of RUTaL, including the expert number N, rank dimension r,

Table 7: Hyperparameter and structure configuration of RUTaL on Pascal Context. Trainable parameters (#TP) and $\Delta\mathbf{m}$(%) are reported.

Experts	#TP	$\Delta\mathbf{m}$
$N=1$	2.38M	+7.61%
$N=4$	2.93M	+7.79%
$N=16$	5.10M	+7.83%
$N=64$	13.79M	+8.07%

(a) Expert Number N

Rank	#TP	$\Delta\mathbf{m}$
$r=1$	2.15M	+7.27%
$r=4$	2.48M	+7.66%
$r=8$	2.93M	+7.79%
$r=16$	3.81M	+7.71%

(b) Rank dimension r

Projection	#TP	$\Delta\mathbf{m}$
$\hat{D}=D/8$	2.26M	+6.96%
$\hat{D}=D/4$	2.93M	+7.79%
$\hat{D}=D/2$	4.77M	+7.95%
$\hat{D}=D$	10.52M	+7.82%

(c) Projection $\hat{D}$

Backbone	#TP	$\Delta\mathbf{m}$
Tiny	2.93M	+7.79%
Small	3.86M	+8.41%
Base	6.02M	+8.54%
Large	11.68M	+12.35%

(d) Backbone Scale

Table 8: Comparison of adaptation strategies on the ViT-based DINOv2 pre-trained backbone. For MTLoRA and TADFormer, their official implementations are adapted to the ViT architecture.

Pre-trained ViT Foundations	**Adaptation Methods**	**SemSeg** (mIoU ↑)	**HumPa** (mIoU ↑)	**Saliency** (mIoU ↑)	**Normals** (RMSE ↓)	$\Delta\mathbf{m}$(%) ↑	#**TP**(M)↓
DINOv2-Reg [50]	Multi-task full	79.74	71.98	65.80	13.94	+15.70	113.68
	Decoder-only	79.32	72.34	61.71	16.79	+10.09	23.63
	MTLoRA [1]	79.42	70.62	65.43	14.17	+14.57	34.15
	TADFormer [3]	78.04	70.12	65.75	14.09	+14.09	33.61
	RUTaL(Ours)	81.83	73.26	64.28	14.42	+15.72	28.98

and the projection dimension $\hat{D}$. As shown in Table 7(a), increasing N from 1 to 4 improves performance (+7.61% to +7.79%) with moderate parameter growth, while larger N (*e.g.*, 64) brings only marginal gains (+8.04%) at substantially higher cost, indicating diminishing returns. We thus adopt $N=4$. Table 7(b) indicates that performance improves as rank r increases from 1 to 8, peaking at $r=8$ (+7.79%), after which performance slightly drops at $r=16$. This suggests that moderate low-rank capacity is sufficient, while excessive rank weakens parameter efficiency. Table 7(c) indicates that performance improves from $D/8$ to $D/2$. However, when $\hat{D}=D$, performance slightly declines due to insufficient training resulting from the increased parameter count. Considering parameter efficiency, we adopt $\hat{D}=D/4$.

Backbone Scaling Effects We evaluate the scalability of RUTaL across backbone capacity, pretraining scale, and architecture. Table 7(d) shows that as the Swin Transformer backbone scales from Tiny to Large, performance consistently improves. This indicates that RUTaL effectively adapts to models of varying scales and achieves larger gains with increased capacity. In Table 8, we further adopt a vanilla ViT backbone and evaluate the adaptation performance of different methods on the advanced vision foundation model (VFM) DINOv2 [50]. The results show that our method consistently outperforms conventional multi-task full parameter fine-tuning and decoder-only fine-tuning as well as advanced multi-task PEFT methods MTLoRA and TADFormer, while achieving superior parameter efficiency. This demonstrates that RUTaL can generalize effectively across different Transformer backbones. Fig. 3 shows that larger pretraining

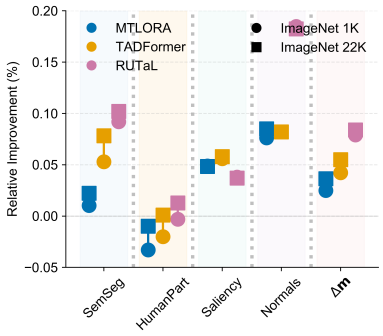

Fig. 3: Performance of adaptation methods with backbones pre-trained on datasets of different scales.

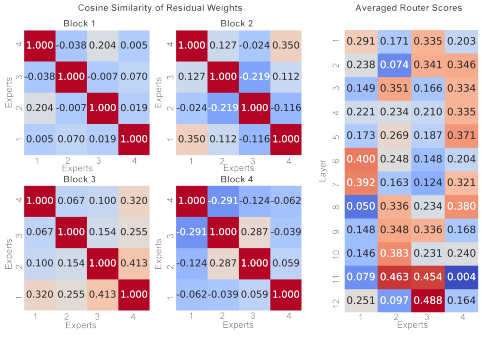

Fig. 4: Expert residual weight similarity in the last layer of four backbone stage and the average router scores across all layers.

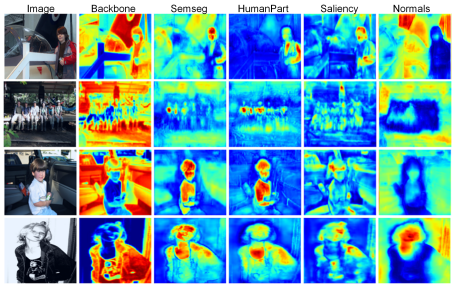

Fig. 5: Visualization of features learned by the task-generic backbone and the task-specific side branch.

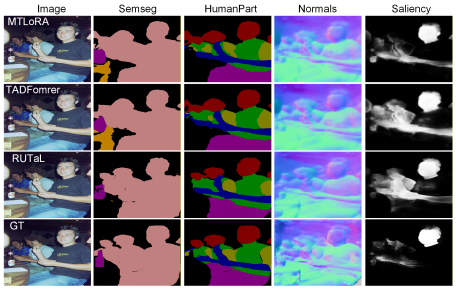

Fig. 6: Qualitative comparison examples of MTLoRA, TADFormer, and the proposed RUTaL.

datasets lead to stronger adaptation performance, while RUTaL consistently outperforms MTLoRA and TADFormer across different pretraining scales.

Visualizations We further conduct visualization to provide deeper insights into the mechanism of RUTaL. Fig. 4 (left) shows the cosine similarity matrix of the four residual experts in the last layer of each block. The low inter-expert similarity indicates that TGRU learns diverse and specialized experts. Fig. 4 (right) presents the averaged routing scores across layers, which exhibit a balanced distribution, demonstrating effective load balancing without expert collapse.

To analyze feature learning behavior, we visualize task-shared and task-specific representations in Fig. 5. The shared features capture discriminative semantic cues across object parts, while the branch features exhibit clear task specificity—for example, focusing on body regions in human part segmentation and emphasizing structural foreground–background cues in surface normal estimation. These results validate the effectiveness of the proposed decoupled adaptation mechanism. Fig. 6 further presents qualitative comparisons, showing that our method produces more reliable predictions.

5 Conclusion

RUTaL, a decoupled PEFT framework for MTL, is proposed in this paper. We revisit the coupled design principle of existing PEFT approaches for MTL and design a decoupled framework that separates task-generic backbone capacity expansion from task-specific specialization. Specifically, Task-General Residual Upcycling (TGRU) restructures a pre-trained backbone into a task-generic Mixture-of-Experts (MoE) architecture to establish a high-capacity shared foundation, while Task-Specific Ladder Adaptation (TSLA) derives lightweight task-specialized representations through a decoupled pathway. Extensive experiments demonstrate that RUTaL achieves state-of-the-art multi-task dense prediction performance with fewer trainable parameters and improved computational efficiency. More broadly, this decoupled perspective offers a principled direction for other MTL settings, such as continual task adaptation. In future work, we plan to explore the application of multimodal models to multi-task learning beyond purely visual domains.

References

1. Agiza, A., Neseem, M., Reda, S.: Mtlora: Low-rank adaptation approach for efficient multi-task learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16196–16205 (2024)
2. Arampatzakis, V., Pavlidis, G., Mitianoudis, N., Papamarkos, N.: Monocular depth estimation: A thorough review. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(4), 2396–2414 (2023)
3. Baek, S., Lee, S., Jo, H., Choi, H., Min, D.: Tadformer: Task-adaptive dynamic transformer for efficient multi-task learning. arXiv preprint arXiv:2501.04293 (2025)
4. Brüggemann, D., Kanakis, M., Obukhov, A., Georgoulis, S., Van Gool, L.: Exploring relational context for multi-task dense prediction. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021)
5. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
6. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
7. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. In: International Conference on Learning Representations (ICLR) (2023)
8. Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L.: Dynamic detr: End-to-end object detection with dynamic attention. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2988–2997 (2021)
9. Diao, H., Wan, B., Zhang, Y., Jia, X., Lu, H., Chen, L.: Unipt: Universal parallel tuning for transfer learning with efficient parameter and memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28729–28740 (2024)
10. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.M., Chen, W., et al.: Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence **5**(3), 220–235 (2023)

11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
12. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research **23**(120), 1–39 (2022)
13. Feng, Y., Shen, B., Gu, N., Zhao, J., Fu, P., Lin, Z., Wang, W.: Dive into moe: Diversity-enhanced reconstruction of large language models from dense into mixture-of-experts. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 19375–19394 (2025)
14. Fu, M., Zhu, K., Ding, Z., Wu, J.: Dtl: Parameter-and memory-efficient disentangled vision learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
15. Fu, Z., Yang, H., So, A.M.C., Lam, W., Bing, L., Collier, N.: On the effectiveness of parameter-efficient fine-tuning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 12799–12807 (2023)
16. Han, C., Wang, Q., Cui, Y., Cao, Z., Wang, W., Qi, S., Liu, D.: Eˆ 2vpt: An effective and efficient approach for visual prompt tuning. arXiv preprint arXiv:2307.13770 (2023)
17. Han, G., Lim, S.N.: Few-shot object detection with foundation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28608–28618 (2024)
18. He, E., Khattar, A., Prenger, R., Korthikanti, V., Yan, Z., Liu, T., Fan, S., Aithal, A., Shoeybi, M., Catanzaro, B.: Upcycling large language models into mixture of experts. arXiv preprint arXiv:2410.07524 (2024)
19. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022)
20. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=0RDcd5Axok
21. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 2790–2799. PMLR (2019)
22. Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022)
23. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
24. Huang, H., Huang, Y., Lin, L., Tong, R., Chen, Y.W., Zheng, H., Li, Y., Zheng, Y.: Going beyond multi-task dense prediction with synergy embedding models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28181–28190 (2024)
25. Huang, Y., Ye, P., Huang, C., Cao, J., Zhang, L., Li, B., Yu, G., Chen, T.: Ders: Towards extremely efficient upcycled mixture-of-experts models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10056–10066 (2025)
26. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European conference on computer vision. pp. 709–727. Springer (2022)

27. Jiang, F., Wang, S., Gong, X.: Task-conditional adapter for multi-task dense prediction. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 2059–2068 (2024)
28. Jie, S., Deng, Z.H.: Convolutional bypasses are better vision transformer adapters. arXiv preprint arXiv:2207.07039 (2022)
29. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
30. Karimi Mahabadi, R., Henderson, J., Ruder, S.: Compacter: Efficient low-rank hypercomplex adapter layers. Advances in Neural Information Processing Systems (NeurIPS) **34**, 1022–1035 (2021)
31. Karimi Mahabadi, R., Ruder, S., Dehghani, M., Henderson, J.: Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (2021)
32. Kerssies, T., Cavagnero, N., Hermans, A., Norouzi, N., Averta, G., Leibe, B., Dubbelman, G., De Geus, D.: Your vit is secretly an image segmentation model. In: Proceedings of the computer vision and pattern recognition conference. pp. 25303–25313 (2025)
33. Komatsuzaki, A., Puigcerver, J., Lee-Thorp, J., Ruiz, C.R., Mustafa, B., Ainslie, J., Tay, Y., Dehghani, M., Houlsby, N.: Sparse upcycling: Training mixture-of-experts from dense checkpoints. arXiv preprint arXiv:2212.05055 (2022)
34. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3045–3059 (2021)
35. Li, J., Wang, X., Zhu, S., Kuo, C.W., Xu, L., Chen, F., Jain, J., Shi, H., Wen, L.: Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. Advances in Neural Information Processing Systems **37**, 131224–131246 (2024)
36. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)
37. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (2021)
38. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)
39. Liu, B., Liu, X., Jin, X., Stone, P., Liu, Q.: Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems **34**, 18878–18890 (2021)
40. Liu, Y.C., Ma, C.Y., Tian, J., He, Z., Kira, Z.: Polyhistor: Parameter-efficient multi-task adaptation for dense vision tasks. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2021)
42. Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., Feris, R.: Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5334–5343 (2017)

43. Lu, Y., Cao, S., Wang, Y.X.: Swiss army knife: Synergizing biases in knowledge from vision foundation models for multi-task learning. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=eePww5u7J3
44. Luo, W., Cao, Y., Yao, H., Zhang, X., Lou, J., Cheng, Y., Shen, W., Yu, W.: Exploring intrinsic normal prototypes within a single image for universal anomaly detection. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 9974–9983 (2025)
45. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1851–1860 (2019)
46. Mantri, K.S.I., Schönlieb, C.B., Ribeiro, B., Baskin, C., Eliasof, M.: Ditask: Multi-task fine-tuning with diffeomorphic transformations. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 25218–25229 (2025)
47. Mercea, O.B., Gritsenko, A., Schmid, C., Arnab, A.: Time-memory-and parameter-efficient visual adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5536–5545 (2024)
48. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
49. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 891–898 (2014)
50. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
51. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
52. Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., Gurevych, I.: Adapterhub: A framework for adapting transformers. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020)
53. Ruder, S., Bingel, J., Augenstein, I., Søgaard, A.: Latent multi-task architecture learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 4822–4829 (2019)
54. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
55. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 746–760. Springer (2012)
56. Sung, Y.L., Cho, J., Bansal, M.: Lst: Ladder side-tuning for parameter and memory efficient transfer learning. Advances in Neural Information Processing Systems **35**, 12991–13005 (2022)
57. Tang, A., Shen, L., Luo, Y., Yin, N., Zhang, L., Tao, D.: Merging multi-task models via weight-ensembling mixture of experts. In: Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 47778–47799. PMLR (21–27 Jul 2024), https://proceedings.mlr.press/v235/tang24e.html

58. Tang, A., Shen, L., Luo, Y., Zhan, Y., Hu, H., Du, B., Chen, Y., Tao, D.: Parameter efficient multi-task model fusion with partial linearization. arXiv preprint arXiv:2310.04742 (2023)
59. Vandenhende, S., Georgoulis, S., Van Gool, L.: Mti-net: Multi-scale task interaction networks for multi-task learning. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part IV 16. pp. 527–543. Springer (2020)
60. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3349–3364 (2020)
61. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
62. Xin, Y., Du, J., Wang, Q., Lin, Z., Yan, K.: Vmt-adapter: Parameter-efficient transfer learning for multi-task dense scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 38, pp. 16085–16093 (2024)
63. Xin, Y., Yang, J., Luo, S., Du, Y., Qin, Q., Cen, K., He, Y., Zhang, Z., Fu, B., Yang, X., et al.: Parameter-efficient fine-tuning for pre-trained vision models: A survey and benchmark. arXiv preprint arXiv:2402.02242 (2024)
64. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2945–2954 (2023)
65. Xu, Y., Li, X., Yuan, H., Yang, Y., Zhang, L.: Multi-task learning with multi-query transformer for dense prediction. IEEE Transactions on Circuits and Systems for Video Technology (2023)
66. Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., Tao, D.: Adamerging: Adaptive model merging for multi-task learning. arXiv preprint arXiv:2310.02575 (2023)
67. Yang, Y., Jiang, P.T., Hou, Q., Zhang, H., Chen, J., Li, B.: Multi-task dense prediction via mixture of low-rank experts. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 27927–27937 (2024)
68. Yang, Y., Jiang, P.T., Hou, Q., Zhang, H., Chen, J., Li, B.: Multi-task dense predictions via unleashing the power of diffusion. In: Yue, Y., Garg, A., Peng, N., Sha, F., Yu, R. (eds.) International Conference on Learning Representations. vol. 2025, pp. 18813–18830 (2025), https://proceedings.iclr.cc/paper_files/paper/2025/file/2f2b1d6bbd50865eca40e2774a057eef-Paper-Conference.pdf
69. Ye, H., Xu, D.: Invpt++: Inverted pyramid multi-task transformer for visual scene understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
70. Yi-Lin Sung, Jaemin Cho, M.B.: Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
71. Yin, D., Han, X., Li, B., Feng, H., Bai, J.: Parameter-efficient is not sufficient: Exploring parameter, memory, and time efficient adapter tuning for dense predictions. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 1398–1406 (2024)
72. Yu, B.X., Chang, J., Liu, L., Tian, Q., Chen, C.W.: Towards a unified view on visual parameter-efficient transfer learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022)

73. Yu, J., Dai, Y., Liu, X., Huang, J., Shen, Y., Zhang, K., Zhou, R., Adhikarla, E., Ye, W., Liu, Y., et al.: Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras. arXiv preprint arXiv:2404.18961 (2024)
74. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. Advances in neural information processing systems **33**, 5824–5836 (2020)
75. Zaken, E.B., Goldberg, Y., Ravfogel, S.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 1–9 (2022)
76. Zaken, E.B., Goldberg, Y., Ravfogel, S.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (2022)
77. Zhang, J.O., Sax, A., Zamir, A., Guibas, L., Malik, J.: Side-tuning: a baseline for network adaptation via additive side networks. In: European conference on computer vision. pp. 698–714. Springer (2020)
78. Zhang, J., Fan, J., Ye, P., Zhang, B., Ye, H., Li, B., Cai, Y., Chen, T.: Bridgenet: Comprehensive and effective feature interactions via bridge feature for multi-task dense predictions. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
79. Zhang, K., Zhou, R., Adhikarla, E., Yan, Z., Liu, Y., Yu, J., Liu, Z., Chen, X., Davison, B.D., Ren, H., et al.: A generalist vision–language foundation model for diverse biomedical tasks. Nature Medicine **30**(11), 3129–3141 (2024)
80. Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE transactions on knowledge and data engineering **34**(12), 5586–5609 (2021)
81. Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J.: Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4106–4115 (2019)
82. Zhou, X., Liang, D., Xu, W., Zhu, X., Xu, Y., Zou, Z., Bai, X.: Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14707–14717 (2024)
83. Zhu, T., Qu, X., Dong, D., Ruan, J., Tong, J., He, C., Cheng, Y.: Llama-moe: Building mixture-of-experts from llama with continual pre-training. In: Proceedings of the 2024 conference on empirical methods in natural language processing. pp. 15913–15923 (2024)
84. Zhu, X., Guan, Y., Liang, D., Chen, Y., Liu, Y., Bai, X.: Moe jetpack: From dense checkpoints to adaptive mixture of experts for vision tasks. Advances in Neural Information Processing Systems **37**, 12094–12118 (2024)
85. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)