Resolving Token-Space Gradient Conflicts: Token Space Manipulation for Transformer-Based Multi-Task Learning

Wooseong Jeong
KAIST
stk14570@kaist.ac.kr

Kuk-Jin Yoon
KAIST
kjyoon@kaist.ac.kr

Abstract

Multi-Task Learning (MTL) enables multiple tasks to be learned within a shared network, but differences in objectives across tasks can cause negative transfer, where the learning of one task degrades another task's performance. While pre-trained transformers significantly improve MTL performance, their fixed network capacity and rigid structure limit adaptability. Previous dynamic network architectures attempt to address this but are inefficient as they directly convert shared parameters into task-specific ones. We propose Dynamic Token Modulation and Expansion (DTME-MTL), a framework applicable to any transformer-based MTL architecture. DTME-MTL enhances adaptability and reduces overfitting by identifying gradient conflicts in token space and applying adaptive solutions based on conflict type. Unlike prior methods that mitigate negative transfer by duplicating network parameters, DTME-MTL operates entirely in token space, enabling efficient adaptation without excessive parameter growth. Extensive experiments demonstrate that DTME-MTL consistently improves multi-task performance with minimal computational overhead, offering a scalable and effective solution for enhancing transformer-based MTL models.

1. Introduction

Multi-Task Learning (MTL) enables multiple tasks to be learned simultaneously within a shared network, improving generalization, efficiency, and convergence speed compared to training separate models [6]. However, conflicting objectives among tasks can lead to *negative transfer*, where learning one task degrades the performance of another [10]. The key challenge lies in designing architectures that effectively balance shared and task-specific representations to mitigate negative transfer.

Transformer-based MTL architectures [63, 64, 67] leverage the strong generalization capabilities of large-scale pre-trained networks such as Vision Transformers (ViTs) [14]. By utilizing pre-trained transformers trained on large open-source datasets, these architectures demonstrate improved generalization compared to conventional CNN-based MTL methods [11, 16, 41, 53, 57, 61, 71, 72]. However, they typically rely on predefined modules, such as Task Prompter mechanisms [63, 64, 67] and Mixture of Experts (MoE) [9, 18, 44, 48, 70], to separate shared and task-specific components. These rigid structures struggle to adapt to dynamic task relationships, leading to inefficient information sharing and suboptimal performance. The degree of task specialization required varies across different network depths [15]: high-level tasks such as semantic segmentation demand greater capacity in deeper layers, while low-level tasks like surface normal estimation rely more on shallower layers. Ideally, MTL architectures should dynamically adjust the allocation of shared and task-specific representations to accommodate these variations. However, existing transformer-based MTL frameworks are inherently constrained by their fixed network capacity, limiting their ability to adapt to evolving task dependencies and effectively mitigate negative transfer.

A straightforward approach to addressing these limitations is to increase the size of the transformer backbone. While this expands network capacity, it has a major drawback: it prevents the use of open-source pre-trained networks, which provide strong initialization and generalization capabilities across multiple tasks. Training a larger network from scratch requires massive computational resources and large-scale datasets, making this approach impractical for many applications. Instead, an effective MTL framework should refine existing architectures to retain the advantages of pre-trained transformers while improving adaptability to task-specific needs.

To achieve this, efficient adaptation methods for pre-trained transformer-based MTL architectures are needed. Unlike approaches that build MTL frameworks from scratch or rely on task optimization within a fixed network capacity, we focus on adapting and enhancing predefined architec-

*Our source code is available at: https://github.com/wooseong97/DTME-MTL tures while preserving their core design. This allows existing MTL models to be improved efficiently while dynamically adjusting task-specific representations. Despite its potential, the challenge of how to adaptively expand existing multi-task networks remains an underexplored problem.

One possible approach for adapting models during fine-tuning is the use of multi-task optimization techniques [13, 23, 33, 35–38, 45, 49, 50, 68], which mitigate negative transfer by adjusting task loss weights or modifying gradients. While these methods help balance task performance, they remain limited by a fixed network design and cannot expand model capacity.

A more direct approach involves dynamic network architectures, such as Recon [22], which directly expand network capacity to mitigate negative transfer. Recon measures conflicting gradients [68] in each layer—where gradients from different tasks point in opposing directions—and transforms the most conflicting layers into task-specific ones. While this increases flexibility by expanding the capacity of predefined architectures, directly converting shared parameters into task-specific ones in transformers leads to parameter inefficiency, excessive computational overhead, and a higher risk of overfitting. Consequently, its scalability to large transformer-based architectures is limited.

To address these challenges, we propose **Dynamic Token Modulation and Expansion (DTME-MTL)**, a novel framework designed to improve pre-trained transformer-based MTL architectures. Unlike previous methods that directly manipulate network parameters, our approach mitigates negative transfer by restructuring the *token space* of multi-task networks. We treat transformer tokens as learnable parameters and analyze their structure using *singular value decomposition* (SVD) to identify gradient conflicts in token space. These conflicts are categorized into two types: *range space conflicts*, which are addressed through modulation via affine transformation of existing tokens, and *null space conflicts*, which are resolved by introducing new task-specific tokens through expansion.

In our experiments, we demonstrate that DTME-MTL effectively enhances multi-task performance with minimal parameter overhead. Additionally, our results highlight that resolving task conflicts in the token space improves adaptability while mitigating overfitting.

Our main contributions are summarized as follows:

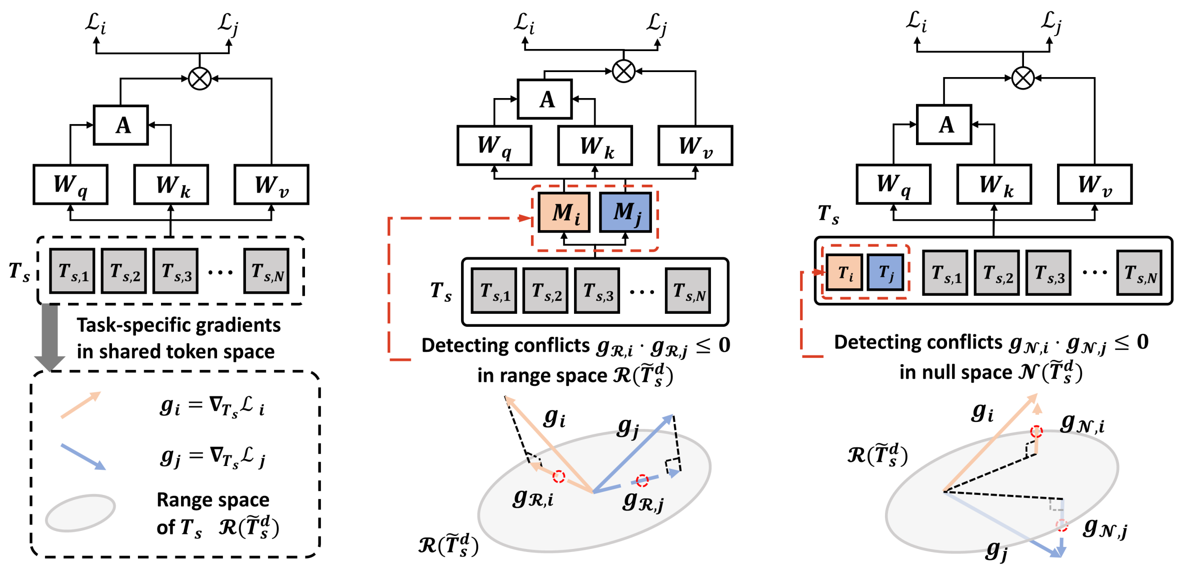
- We propose DTME-MTL which dynamically modulates and expands token spaces to mitigate negative transfer in transformer-based multi-task architectures.
- We introduce a structured approach to resolving gradient conflicts in token space by categorizing them into range and null space conflicts, demonstrating how this improves multi-task performance.
- DTME-MTL is an *off-the-shelf* solution that seamlessly integrates with existing state-of-the-art transformer-based MTL architectures, enhancing performance with minimal computational overhead.

2. Related Works

Multi-Task Learning in Vision Transformers. Originally designed for NLP tasks, transformers have outperformed existing CNN models in various computer vision tasks. Attempts have been made to incorporate Vision Transformer [14, 39, 58–60, 65] in MTL. MTFormer [62] employs a shared transformer encoder and decoder with a cross-task attention mechanism. MulT [1] utilizes a shared attention mechanism to model task dependencies based on the Swin transformer. InvPT [66] focuses on global spatial position and multi-task context for dense prediction tasks through multi-scale feature aggregation. Mixture of Experts (MoE), inspired by the NLP domain, divides the model into predefined expert groups, adaptively shared or devoted to specific tasks during the learning phase [9, 18, 27, 44, 48, 70]. Task prompter [63, 64, 67] uses task-specific tokens to encapsulate task-specific information and employs cross-task interactions to enhance multi-task performance.

Dynamic Network Architectures for MTL. Dynamic networks adapt their structure during training to improve efficiency and task specialization. Several methods have explored dynamic architectures for MTL. Channel-wise dynamic allocation [2] assigns different convolutional channels to different tasks, but this method is not directly applicable to transformer-based architectures. Neural Architecture Search (NAS) for MTL [3, 5, 21, 24, 34, 47, 52] explores optimal network configurations but is computationally expensive and incompatible with large pre-trained backbone models such as ViTs [14]. Recon [22] transforms shared parameters directly into task-specific ones to handle conflicting gradients. Unlike most dynamic network architectures, our approach focuses on a dynamic system that can be directly applied to transformer-based multi-task architectures, leveraging pre-trained backbones while maintaining a reasonable computational cost.

Multi-Task Optimization. Optimizing the MTL aims to address negative transfer by adjusting the relative weighting of task losses or directly manipulating gradients. Task-dependent uncertainty [33] is utilized to weigh the loss of multiple tasks. Liu et al. [38] considers the rate of loss descent, while [23] prioritizes tasks based on difficulty. Recently, Liu et al. [36] proposed updating task weights based on the loss history. In contrast, approaches like [13, 35, 37, 45, 49, 50, 68] directly modify task gradients to achieve the desired balance. PCGrad [68] analyzes negative transfer by identifying conflicting gradients in the shared parameters of the network. Jiang et al. [31] suggests a positive link between negative transfer and conflicting gradients in auxiliary task learning. However, the conventional view



(a) SVD in shared token space **(b) Token modulation by detecting range space gradient conflicts** **(c) Token expansion by detecting null space gradient conflicts**

Figure 1. Framework overview of the proposed DTME-MTL. (a) At each network layer, we compute the input token's range space $\mathcal{R}(\widetilde{\mathcal{T}}_s^d)$ and their task-specific gradients, determining principal vectors from the uncentered covariance of $\mathcal{T}_s$. (b) In cases where task-specific gradients conflict in the range space of $\widetilde{\mathcal{T}}_s^d$ (e.g. $g_{\mathcal{R},i} \cdot g_{\mathcal{R},j} \leq 0$), modulation is applied to $\mathcal{T}_s$ by introducing $\mathcal{M}_i$ and $\mathcal{M}_j$. (c) When task-specific gradients conflict within the null space of $\widetilde{\mathcal{T}}_s^d$ (e.g. $g_{\mathcal{N},i} \cdot g_{\mathcal{N},j} \leq 0$), task-specific tokens $\mathcal{T}_i$ and $\mathcal{T}_j$ are added.

in MTL considers conflicting gradients a key factor contributing to negative transfer in joint multi-task learning optimization [13, 29, 30, 35, 37, 45, 49, 50, 68], where tasks are learned together rather than serving as auxiliary tasks. Therefore, we adopt a similar perspective. Normalized gradients are employed to prevent spillover between tasks [7], whereas Chen et al. [8] introduce stochasticity to the network's parameters based on the consistency in the sign of gradients. RotoGrad [28] rotates the feature space of the network to narrow the gap between tasks.

3. Preliminaries

In multi-task learning, the network learns a set of tasks $\{\tau_i\}_{i=1}^{\mathcal{K}}$ jointly, where $\mathcal{K}$ is the number of tasks. Each task τ_i has its own loss function $\mathcal{L}_i$. The network parameter Θ can be classified into $\Theta = \{\Theta_s, \Theta_1, ..., \Theta_\mathcal{K}\}$ where Θ_s is shared parameter across all tasks and Θ_i is task-specific parameters devoted to task τ_i. Then, the objective function of multi-task learning is to minimize the weighted sum of all tasks' losses: $\Theta^* = \arg\min_\Theta \sum_{i=1}^{\mathcal{K}} w_i \mathcal{L}_i(\Theta_s, \Theta_i)$ where w_i represents the scale of the task-specific loss $\mathcal{L}_i$. A phenomenon called conflicting gradients [68], where the gradients of each objective point in different directions, has been identified as a main cause of negative transfer.

Definition 1 (Conflicting gradients)**.** *Define g_i as the gradient of task τ_i with respect to the shared parameters Θ_s as $g_i = \nabla_{\Theta_s} \mathcal{L}_i(\Theta_s, \Theta_i)$. Let g_i and g_j represent the gradients for a pair of tasks τ_i and τ_j where $i \neq j$. If $g_i \cdot g_j \leq 0$, these two gradients are termed conflicting gradients.*

However, the role of conflicting gradients remains a topic of debate. While conventional MTL optimization studies [13, 29, 35, 37, 45, 49, 50, 68] consider conflicting gradients as a main cause of negative transfer, Jiang et al. [31] argue that they can serve as a form of regularization that improves generalization when present in network parameters. Our findings align with Jiang et al. [31] in that directly resolving conflicting gradients by converting shared parameters into task-specific ones [22] leads to overfitting when applied to transformers. In contrast, we propose a token-based network expansion approach that categorizes gradient conflicts within token space and adapts accordingly, mitigating negative transfer while maintaining generalization.

4. Method

In order to mitigate negative transfer by ensuring sufficient space for tasks, we adopt token-based network expansion. Initially, we define the token space as the output of each layer in the transformer block through singular value decomposition (SVD). Subsequently, we categorize conflicts in task-specific gradients into two types: conflicts in the range space of tokens and conflicts in the null space of

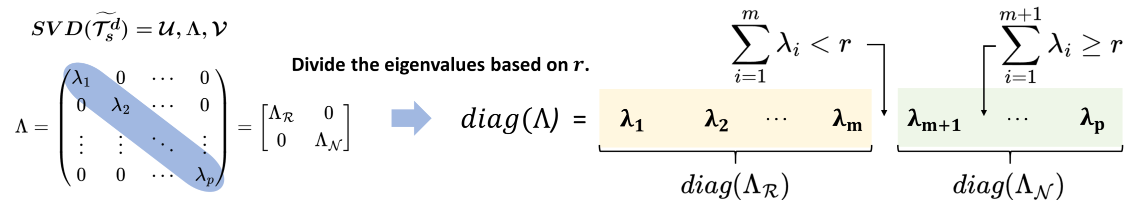


Figure 2. The process approximates the range and null spaces of $\widetilde{\mathcal{T}}_s^d$ based on the proportion of total variance, r. These eigenvalues are arranged in descending order, satisfying $\lambda_i \geq \lambda_j$ if $i < j$. If r is greater than the sum up to λ_m and smaller than the sum up to λ_{m+1}, then we select the set $\{\lambda_i\}_{i=1}^{m}$ as $\Lambda_{\mathcal{R}}$, and the remaining set $\{\lambda_i\}_{i=m+1}^{p}$ as $\Lambda_{\mathcal{N}}$.

tokens. Finally, based on the type of conflict, we introduce efficient token modulation and expansion techniques for transformer-based multi-task architectures.

4.1. Defining Token Space using SVD

In this section, we create a vector space consisting of shared tokens in a transformer, aiming to classify the types of conflicting gradients. More specifically, we approximate the range space and null space of the uncentered covariance of the tokens before applying our methods.

Let's consider a dataset $\{\mathcal{X}_l, \mathcal{Y}_l\}_{l=1}^{n}$, where $\mathcal{X}_l$ represents the input, $\mathcal{Y}_l$ denotes the label, and n is the number of samples. Denote input shared token for a layer d as $\mathcal{T}_s^{l,d} = [\mathcal{T}_{s,1}^{l,d}, \mathcal{T}_{s,2}^{l,d}, ..., \mathcal{T}_{s,N}^{l,d}]$ where N is the total number of shared tokens in that layer. Every token $\mathcal{T}_{s,k}^{l,d} \in \mathbb{R}^p$ represents the output of the transformer layer $d-1$ for the corresponding input data $\mathcal{X}_l$, where p is the hidden dimension of the token embedding. Let's consider a total of D transformer layers. Next, the uncentered covariance of the token in layer d (where $1 \leq d \leq D$) is as follows:

$$\widetilde{\mathcal{T}}_s^d = \frac{1}{n} \sum_{l=1}^{n} (\mathcal{T}_s^{l,d})(\mathcal{T}_s^{l,d})^T \tag{1}$$

$\widetilde{\mathcal{T}}_s^d$ is a square matrix of dimensions $p \times p$. To define the token space, we apply SVD to $\widetilde{\mathcal{T}}_s^d$. Following this, we can divide vector space formed by $\widetilde{\mathcal{T}}_s^d$ into its range space $\mathcal{R}(\widetilde{\mathcal{T}}_s^d)$ and null space $\mathcal{N}(\widetilde{\mathcal{T}}_s^d)$ depending on the magnitude of eigenvalue Λ. The process is illustrated below:

$$\widetilde{\mathcal{T}}_s^d = \mathcal{U}\Lambda\mathcal{V}^T, \quad \Lambda = \begin{bmatrix} \Lambda_{\mathcal{R}} & 0 \\ 0 & \Lambda_{\mathcal{N}} \end{bmatrix} \tag{2}$$

where Λ is a diagonal matrix. Each $\Lambda_{\mathcal{R}}$ and $\Lambda_{\mathcal{N}}$ represent submatrices of Λ containing the eigenvalues of the range space and null space, respectively. Both $\mathcal{U}$ and $\mathcal{V}$ are square matrices, each with dimensions $p \times p$.

From Eq. (2), we obtain a mathematical tool to define the range and null space of the covariance of the token, $\widetilde{\mathcal{T}}_s^d$. To approximate the range space, we choose the eigenvalue $\Lambda_{\mathcal{R}}$ along with their corresponding eigenvectors from $\mathcal{U}_{\mathcal{R}}$. On the other hand, when approximating the null space, we should select the eigenvalues $\Lambda_{\mathcal{N}}$ and their corresponding eigenvectors from $\mathcal{U}_{\mathcal{N}}$. Ideally, we should choose eigenvalues that are exactly zero to form the null space. However, in practice, Λ can not be precisely zero. Therefore, it's essential to establish a criterion for selecting the eigenvalue to distinguish between these two spaces.

Instead of introducing a new manually designed rule for approximating each range and null space of $\widetilde{\mathcal{T}}_s^d$, we opt to directly employ the evaluation tool for the SVD [32] as criteria for determining the range and null space of tokens. In assessing the accuracy of the SVD approximation, the proportion of total variance, denoted as r, has been employed:

$$r = \frac{\sum_{\lambda \in diag(\Lambda_{\mathcal{N}})} \lambda}{\sum_{\lambda \in diag(\Lambda_{\mathcal{R}})} \lambda} \tag{3}$$

The $diag$ function serves as an operator, returning a set containing the diagonal entries of the input matrix. In our approach, we employ Eq. (3) to directly divide the range and null space of $\widetilde{\mathcal{T}}_s^d$. As depicted in Fig. 2, the diagonal elements of the matrix Λ, obtained through the SVD of $\widetilde{\mathcal{T}}_s^d$, are arranged in descending order based on their magnitudes. We can select the index of the eigenvalue m such that the sum of eigenvalues up to order m is smaller than r, and the sum up to $m+1$ is larger than r. This index serves as a boundary to divide the range space and null space of $\widetilde{\mathcal{T}}_s^d$.

4.2. Types of Gradient Conflicts

In Section 4.1, we create a p-dimensional vector space using the uncentered covariance of the shared token $\widetilde{\mathcal{T}}_s^d$, linked to the input data set $\{\mathcal{X}\}_{l=1}^{n}$. This vector space is divided into the range and null space, with each space spanned by eigenvectors corresponding to singular values selected based on a specified ratio r. In the upcoming sections, we pinpoint the types of gradient conflict within the vector space we've constructed. We then address these conflicts adaptively by introducing token modulation and expansion techniques.

Using Eq. (2) and Eq. (3), we can partition the eigenvectors of the p-dimensional vector space into its range and null space. Now, let's consider the shared tokens $\mathcal{T}_s^{l,d} = [\mathcal{T}_{s,1}^{l,d}, \ldots, \mathcal{T}_{s,N}^{l,d}]$, omitting the explicit notation of l, d for simplicity. For example, we write $\mathcal{T}_s^{l,d} \rightarrow \mathcal{T}_s$, $\mathcal{T}_{s,k}^{l,d} \rightarrow \mathcal{T}_{s,k}$, and $\tilde{\mathcal{T}}_s^d \rightarrow \tilde{\mathcal{T}}_s$. We treat $\mathcal{T}_s$ as network parameters, for which gradients can be computed during the backpropagation process. Then, for each loss $\mathcal{L}_i$, the task-specific gradient for $\mathcal{T}_{s,k}$ is denoted as $g_i = \nabla_{\mathcal{T}_{s,k}} \mathcal{L}_i$. Consequently, we obtain task-specific gradients $\{g_i\}_{i=1}^{\mathcal{K}}$ corresponding to a set of losses $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ for $\mathcal{T}_s$ as shown in Fig. 1-(a).

Each task-specific gradient g_i can be decomposed into two components, $g_{\mathcal{R},i}$ and $g_{\mathcal{N},i}$, through projection onto the range and null space of $\tilde{\mathcal{T}}_s^d$, respectively. This breakdown is expressed as follows:

$$g_{\mathcal{R},i} = (\mathcal{U}_{\mathcal{R}} \mathcal{U}_{\mathcal{R}}^T) \nabla_{\mathcal{T}_{s,k}} \mathcal{L}_i \quad g_{\mathcal{N},i} = (\mathcal{U}_{\mathcal{N}} \mathcal{U}_{\mathcal{N}}^T) \nabla_{\mathcal{T}_{s,k}} \mathcal{L}_i \tag{4}$$

$\mathcal{U}_{\mathcal{R}}$ and $\mathcal{U}_{\mathcal{N}}$ are orthogonal matrices that consist of eigenvectors of the range space and null space, respectively, with each column representing one eigenvector. Then, the matrices $(\mathcal{U}_{\mathcal{R}} \mathcal{U}_{\mathcal{R}}^T)$ and $(\mathcal{U}_{\mathcal{N}} \mathcal{U}_{\mathcal{N}}^T)$ function as projection operators onto the range and null spaces, respectively.

Building upon the concept of conflicting gradients outlined in Definition 1, we classify conflicts into two types based on the space in which they occur: range space conflicts and null space conflicts. Specifically, conflicts in the range space of tokens occur when $g_{\mathcal{R},i} \cdot g_{\mathcal{R},j} \leq 0$ for any pair of i and j where $i \neq j$. Likewise, conflicts in the null space of tokens emerge when $g_{\mathcal{N},i} \cdot g_{\mathcal{N},j} \leq 0$.

4.3. Token Modulation and Expansion

Building on the gradient conflict types defined in Sec. 4.2, we propose adaptive strategies to mitigate task interference. Specifically, if gradient conflicts occur in the range space, we apply an affine transformation to modulate tokens, while conflicts in the null space are addressed by introducing additional tokens to expand the feature space. This distinction is particularly relevant in the transfer learning setting, where a pretrained transformer backbone is used, and task interference must be handled during fine-tuning. According to [46], training from pretrained weights constrains the model within the same basin of the loss landscape, preserving a feature space similar to that of the pretrained network. This insight guides our separation of token space into range and null spaces: conflicts in the row space indicate that the network already has relevant interpretative capabilities and can be resolved through rotation or scaling, whereas conflicts in the null space suggest the need for additional features, requiring token expansion to enhance the model's capacity.

Token Modulation. In situations where task-specific gradients conflict within the range space of $\tilde{\mathcal{T}}_s$, such as $g_{\mathcal{R},i} \cdot g_{\mathcal{R},j} \leq 0$, modulators $\mathcal{M}_i$ and $\mathcal{M}_j$ are added after the shared token $\mathcal{T}_s$ as shown in Fig. 1-(b). The token modulator $\mathcal{M}$ is a straightforward affine transformation that modulates the shared token $\mathcal{T}_s$ along the channel dimension. To elaborate, considering the embedding dimension of the

---

Algorithm 1: DTME-MTL

Data: Task $\{\tau_i\}_{i=1}^{\mathcal{K}}$, Loss function $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$,
Dataset $\{\mathcal{X}_l, \mathcal{Y}_l\}_{l=1}^{n}$, Shared tokens
$\mathcal{T}_s^{l,d} = \{\mathcal{T}_{s,i}^{l,d}\}_{i=1}^{N}$, Depth of the Network D

for *each layer of the network* ($d \leftarrow 1$ *to* D) **do**
  Get tokens $\{\mathcal{T}_s^{l,d}\}_{l=1}^{n}$ for the layer d corresponding to input $\{\mathcal{X}_l\}_{l=1}^{n}$
  Calculate uncentered covariance. $\tilde{\mathcal{T}}_s^d = \frac{1}{n} \sum_{l=1}^{n} (\mathcal{T}_s^{l,d})(\mathcal{T}_s^{l,d})^T$
  Singular value decomposition. $\mathcal{U}, \Lambda, \mathcal{V} = SVD(\tilde{\mathcal{T}}_s^d)$
  Divide range and null space. $\mathcal{U} = [\mathcal{U}_{\mathcal{R}}, \mathcal{U}_{\mathcal{N}}]$
  Projection to range space. $\{g_{\mathcal{R},i}\}_{i=1}^{\mathcal{K}} = \{(\mathcal{U}_{\mathcal{R}} \mathcal{U}_{\mathcal{R}}^T) \nabla_{\mathcal{T}_{s,k}^{l,d}} \mathcal{L}_i\}_{i=1}^{\mathcal{K}}$
  Projection to null space. $\{g_{\mathcal{N},i}\}_{i=1}^{\mathcal{K}} = \{(\mathcal{U}_{\mathcal{N}} \mathcal{U}_{\mathcal{N}}^T) \nabla_{\mathcal{T}_{s,k}^{l,d}} \mathcal{L}_i\}_{i=1}^{\mathcal{K}}$
  if $g_{\mathcal{R},i} \cdot g_{\mathcal{R},j} \leq 0$ **then**
    Insert token modulators $\mathcal{M}_i$ and $\mathcal{M}_j$
  if $g_{\mathcal{N},i} \cdot g_{\mathcal{N},j} \leq 0$ **then**
    Insert task-specific tokens $\mathcal{T}_i$ and $\mathcal{T}_j$

---

transformer p and the number of shared tokens is N, we can arrange $\mathcal{T}_s$ in the form $[\mathcal{T}_{s,1}, \ldots, \mathcal{T}_{s,N}]$. This arrangement turns $\mathcal{T}_s$ into a $p \times N$ matrix. The modulator $\mathcal{M}$, which incorporates weight and bias $W, b \in \mathbb{R}^p$, performs the transformation $W \odot \mathcal{T}_{s,i} + b$, where $\odot$ denotes elementwise multiplication. When the gradient lies in the row space of $\tilde{\mathcal{T}}_s$, Proposition 1 demonstrates that applying token modulation can effectively resolve gradient conflicts, lowering the multi-task loss.

Proposition 1. *When the input token $\mathcal{T}_{in}$ for input sample $\mathcal{X}_l$ spans the range space of $\tilde{\mathcal{T}}_s$, optimizing the token modulators $\{\mathcal{M}_i\}_{i=1}^{\mathcal{K}}$ reduces gradient conflicts in the row space of $\tilde{\mathcal{T}}_s$ and leads to a reduction in the multi-task loss.*

Token Expansion. Similarly, in cases where task-specific gradients conflict within the null space of $\tilde{\mathcal{T}}_s$, such as $g_{\mathcal{N},i} \cdot g_{\mathcal{N},j} \leq 0$, task-specific tokens $\mathcal{T}_i$ and $\mathcal{T}_j$ are added alongside shared tokens $\mathcal{T}_s$ as shown in Fig. 1-(c). The task-specific tokens $\{\mathcal{T}_i\}_{i=1}^{\mathcal{K}}$ are concatenated with shared tokens before entering the transformer block. Consequently, each task-specific token acquires task-specific information within that layer. Specifically, in a standard transformer block, self-attention is performed for each pair of tokens in the form of $[\mathcal{T}_{s,1}, \ldots, \mathcal{T}_{s,N}] \times [\mathcal{T}_{s,1}, \ldots, \mathcal{T}_{s,N}]$. With token expansion, attention is extended to include $[\mathcal{T}_{s,1}, \ldots, \mathcal{T}_{s,N}] \times [\mathcal{T}_1, \ldots, \mathcal{T}_{\mathcal{K}}]$ on the output. Proposition 2 explains how expanding the token space to address gradi-

Table 1. We conduct an ablation study on dynamic token modulation and expansion, evaluating the multi-task performance of our method on NYUD-v2 and PASCAL-Context. The results of TE, TM, and their combination, TE+TM are presented. We employ a shared encoder and multiple decoders, using ViT-T as the backbone network.

Model	NYUD-v2				PASCAL-Context				
	Semseg mIoU ↑	Depth RMSE ↓	Normal mErr ↓	Edge odsF ↑	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Edge odsF ↑
Baseline (ST)	39.35	0.6611	22.14	59.68	67.96	58.90	83.76	15.65	47.70
Baseline (MT)	34.13	0.6732	22.51	55.30	54.47	51.48	82.04	16.22	41.28
TM	37.85	0.6490	21.75	56.92	64.28	55.10	83.02	15.40	45.80
TE	37.25	0.6553	21.87	57.00	60.51	54.00	82.85	15.55	44.98
TM+TE	38.27	0.6370	21.64	57.90	66.18	56.29	83.41	15.26	47.00
Gain (vs. MT)	**△4.14**	**△0.0362**	**△0.87**	**△2.60**	**△11.71**	**△4.81**	**△1.37**	**△0.96**	**△5.72**
$\triangle_m \uparrow$	**0.044**				**-1.289**				
#Param ↑ (%)	**0.24**				**0.30**				

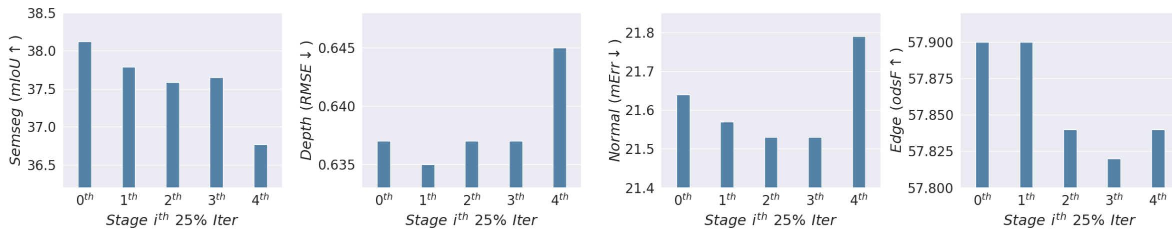


Figure 3. Task performance varies based on when we expand the network. To determine the optimal timing, we assess expansions at the beginning of training and at the end of each quarter iteration, monitoring the corresponding changes in performance.

ent conflicts in the null space of $\tilde{\mathcal{T}}_s$ leads to a reduction in multi-task loss when the gradient lies in this null space. All proofs are provided in Supple E.

Proposition 2. *When the input token $\mathcal{T}_{in}$ for input sample $\mathcal{X}_l$ spans the null space of $\tilde{\mathcal{T}}_s$, token expansion using $\{\mathcal{T}_i\}_{i=1}^{\mathcal{K}}$ alleviates the increase in multi-task loss caused by gradient conflicts in the null space of $\tilde{\mathcal{T}}_s$.*

The complete procedure for DTME-MTL is outlined in Alg.1. Handling gradient conflicts in token space improves adaptability and reduces overfitting while being more efficient than addressing conflicts at the parameter level.

5. Experiments

5.1. Experimental Settings

Datasets and Evaluation. Our method is evaluated on multi-task datasets: NYUD-v2 [51], PASCAL-Context [43] and Taskonomy [69]. Each of them with 4, 5, 11 tasks. To evaluate the performance of tasks, we employed widely used metrics. To evaluate the multi-task performance, we utilize the metric proposed by Maninis et al. [42]. It measures the per-task performance $M_{m,i}$ by averaging it with respect to the single-task baseline $M_{b,i}$, as shown in $\triangle_m = (1/\mathcal{K})\sum_{i=1}^{\mathcal{K}}(-1)^{l_i}(M_{m,i} - M_{b,i})/M_{b,i}$. $l_i = 1$ if a lower value of the measure M_i indicates better performance for task i, and 0 otherwise.

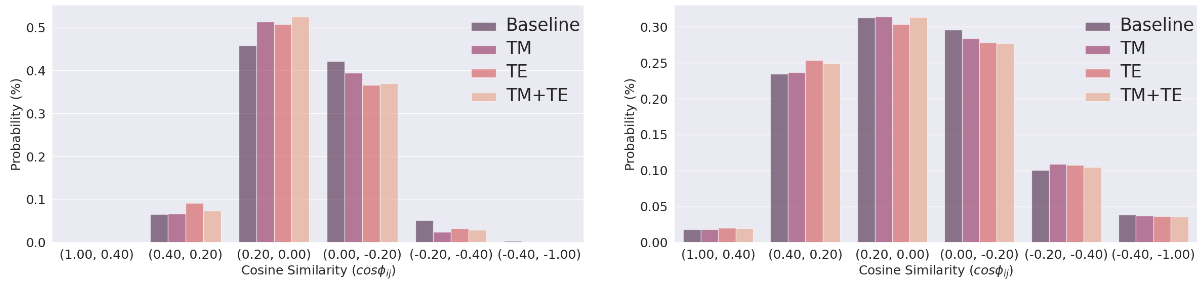
Baselines and Model Variants. For a comprehensive analysis of the proposed DTME-MTL framework, we adopt a typical experimental setup for MTL in our experiments. In Tab. 1, 'Baseline (MT)' refers to a simple multi-task architecture consisting of a shared transformer backbone and basic task-specific decoders. Each decoder comprises one 3×3 Conv-BN-ReLU block. 'Baseline (ST)' has the same structure as 'Baseline (MT)' but is trained with only a single task. We assess the proposed DTME-MTL framework by expanding the network from 'Baseline (MT)' and measure the performance gains achieved by the proposed methods. 'TM' (Token Modulation) signifies the addition of the proposed token modulator to 'baseline (MT)', while 'TE' (Token Expansion) indicates the incorporation of task-specific tokens onto 'Baseline (MT)'. Finally, 'TM+TE' combines both proposed methods. To show how effectively our approach reduces negative transfer, we also compare it with previous multi-task optimization, though our methods can be used alongside them. We include simple gradient descent (GD), GradDrop [8], MGDA [49], PCGrad [68], CAGrad [35], IMTL [37], Nash-MTL [45], and Aligned-MTL [50], as well as loss balancing methods such as UW [33], DWA [38], and FAMO [36]. We also compare our results with dynamic network architecture such as Recon [22]. Further experimental details are summarized in Supple B.

5.2. Experimental Results

Effectiveness of Token Modulation and Expansion. We assess the effectiveness of the proposed methods on the NYUD-v2 and PASCAL-Context datasets, with results detailed in Tab. 1. In the last three rows of the table, we depict the performance gains compared to the two baselines and the increased number of parameters in "$\#Param \uparrow (\%)$". Compared to the Baseline (MT), our methods demonstrate significant performance improvements across all tasks in

Table 2. Performance comparison based on the degree of conflicts in reversed order (Reversed) and randomly selected layers (Random).

Model	NYUD-v2				PASCAL-Context				
	Semseg mIoU ↑	Depth RMSE ↓	Normal mErr ↓	Edge odsF ↑	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Edge odsF ↑
TM+TE	**38.27**	**0.6370**	**21.64**	**57.90**	**66.18**	**56.29**	**83.21**	**15.26**	**47.00**
TM+TE (Random)	36.88	0.6567	22.27	56.30	62.12	54.43	82.95	15.55	45.80
TM+TE (Reverse)	34.71	0.6898	22.59	55.80	57.84	52.82	82.75	15.74	43.20



(a) NYUD-v2

(b) PASCAL-Context

Figure 4. We evaluate the distribution of gradient conflicts by measuring the cosine similarity between task-specific gradients across all shared parameters throughout the optimization process. This is represented as $cos\phi_{ij}$ in (a) for NYUD-v2 and in (b) for PASCAL-Context.

both datasets. Particularly noteworthy is the substantial increase in multi-task performance achieved with just a 0.2% to 0.3% increase in the total network parameters. Additionally, our approach exhibits comparable performance to Baseline (ST) in a multi-task scenario. This implies that reducing negative transfer is achievable by simply integrating token modulators and task tokens, without complex modules.

Analysis of the Timing of Network Expansion. In Fig. 3, we analyze the performance of each task according to the timing of network expansion using the proposed DTME-MTL. Specifically, the timing for expansion refers to the point at which token modulation and expansion are performed based on calculations of the token space using Singular Value Decomposition and measurement of gradient conflicts. The figure illustrates the performance results when network expansion is conducted at the beginning of training (0^{th}) and after each quarter of the entire training process (i^{th} 25% Iter). To ensure fair comparisons, we trained the network using the same number of iterations after the expansion. The results indicate that the optimal timing for expansion may vary across tasks. However, overall, early-stage expansion during network training tends to yield better performance. This aligns with the fact that our approach builds on pre-trained backbone networks.

Analysis of Gradient Conflicts in Parameters. We focus on resolving gradient conflicts in token space. While our primary method operates in token-level representations, we also monitor gradient conflicts in parameter space to better understand training dynamics. In Fig. 4, we visualize the distribution of angles between task-specific gradients of network parameters, categorizing them into different ranges and tracking their frequency over the course of training. When applied to the baseline model, both Token Modulation (TM) and Token Expansion (TE) reduce gradient conflicts in parameter space to some extent. However, as shown in Tab. 7, where our methods are compared with Recon [22], we observe important differences. Recon explicitly suppresses conflicts by modifying network parameters whenever the cosine similarity between task gradients becomes negative. Although this reduces gradient conflicts in parameter space, it often results in severe overfitting and degraded multi-task performance. These findings suggest that conflicts in parameter space are not always reliable indicators of negative transfer. Instead, resolving conflicts in token space offers an alternative strategy that avoids overfitting while still mitigating interference. Additional analysis of token-level conflicts is provided in Appendix D.

Computational Cost of DTME-MTL. In Tab. 6, we report the time consumption for each process of DTME-MTL on PASCAL-Context using a single NVIDIA RTX A6000. We measure the time required for calculating the token space with SVD and for computing gradient conflicts in the token space of the transformer. The time required for each process increases with the size of the transformer. However, the proposed methods are computationally efficient, requiring approximately 1 hour with ViT-L. Considering that typical multi-task architectures require at least one day of training on a single GPU, the computational cost of DTME-MTL is relatively low. Proposed DTME-MTL increases inference time of each task about 13.4% with ViT-B.

Comparing Performance based on Layer Selection Criteria. In Tab. 2, we applied TM+TE to layers with the highest gradient conflicts between tasks. Results are also shown for randomly chosen layers (Random) or layers with the lowest gradient conflicts (Reverse). The network expan-

Table 3. Comparison of multi-task optimization methods on Taskonomy across 11 tasks. Non-converged results are indicated with a dash.

Task Metric	DE L1 Dist. ↓	DZ L1 Dist. ↓	EO L1 Dist. ↓	ET L1 Dist. ↓	Key2D L1 Dist. ↓	Key3D L1 Dist. ↓	N L1 Dist.	PC RMSE ↓	R L1 Dist. ↓	S2D L1 Dist. ↓	S25D L1 Dist. ↓	$\triangle_m$ ↑ (%)
ST	0.0199	0.0195	0.1085	0.1714	0.1633	0.0872	0.2715	0.7586	0.1503	0.1742	0.1504	0.00
GD	0.0187	0.0188	0.1301	0.1757	0.1733	0.0942	0.3076	0.7991	0.1826	0.1902	0.1652	- 7.83
GradDrop [8]	0.0315	0.0242	0.1390	0.1776	0.1778	0.0976	0.4564	0.8644	0.2088	0.1995	0.1752	- 26.11
MGDA [49]	-	-	-	-	-	-	-	-	-	-	-	-
UW [33]	0.0190	0.0190	0.1308	0.1758	0.1734	0.0945	0.3109	0.8009	0.1840	0.1906	0.1657	- 8.43
DWA [38]	0.0186	0.0187	0.1294	0.1759	0.1735	0.0938	0.2788	0.7943	0.1805	0.1902	0.1640	- 6.45
PCGrad [68]	0.0217	0.0192	0.1298	0.1775	0.1714	0.0939	0.2856	0.7985	0.1817	0.1927	0.1595	- 8.29
CAGrad [35]	0.0219	0.0203	0.1314	0.1800	0.1665	0.0932	0.3039	0.8121	0.1874	0.1953	0.1673	- 10.57
IMTL [37]	0.0210	0.0192	0.1282	0.1772	0.1719	0.0936	0.2468	0.7784	0.1734	0.1943	0.1647	- 6.17
Align-MTL [50]	0.0189	0.0193	0.1254	**0.1728**	**0.1664**	0.0914	0.3524	0.8640	0.1938	0.1889	0.1582	- 9.41
Nash-MTL [45]	0.0201	0.0184	0.1248	0.1764	0.1701	0.0921	0.2658	0.7793	0.1706	0.1914	0.1624	- 5.01
FAMO [36]	0.0188	0.0188	0.1300	0.1758	0.1733	0.0942	0.3058	0.7986	0.1826	0.1904	0.1654	- 7.87
DTME-MTL	**0.0150**	**0.0154**	**0.1193**	0.1733	0.1668	**0.0891**	**0.2038**	**0.7373**	**0.1567**	**0.1773**	**0.1517**	+ **4.67**

Table 4. Adaptation of DTME-MTL to other state-of-the-art MTL methods on NYUD-v2.

Task Metric	Semseg mIoU ↑	Depth RMSE ↓	Normal mErr ↓	Edge odsF ↑
MTformer [62]	50.04	0.490	-	-
InvPT [66]	53.56	0.5183	18.81	78.10
+ DTME-MTL	54.38	**0.5020**	18.51	78.20
Taskprompter [67]	55.30	0.5152	18.47	78.20
+ DTME-MTL	**56.36**	0.5122	**18.38**	**78.40**

Table 5. Adaptation of DTME-MTL to other state-of-the-art MTL methods on PASCAL-Context.

Task Metric	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Edge odsF ↑
MTformer [62]	73.51	64.26	67.24	-	-
InvPT [66]	79.03	67.61	84.81	14.15	73.00
+ DTME-MTL	**81.91**	**71.13**	**84.96**	13.73	**73.80**
Taskprompter [67]	80.89	68.89	84.83	13.72	73.50
+ DTME-MTL	81.01	69.08	84.75	**13.65**	73.60

sion system, using conflict detection, outperforms random selection across all tasks. Particularly, applying TM+TE to layers with severe conflict levels consistently outperforms its application in layers with lower conflict levels, validating the effectiveness of the strategy.

Comparison with Multi-Task Optimization. In Tab. 3, we compare DTME-MTL with previous multi-task optimization approaches to demonstrate its effectiveness in reducing negative transfer between tasks on the Taskonomy benchmark using ViT-B. DTME-MTL achieves the best multi-task performance, improving each task by an average of 4.67% with only a 0.118% increase in the number of parameters. Although DTME-MTL introduces additional parameters to address negative transfer, making direct comparisons with optimization methods less straightforward, it consistently improves multi-task performance. However, using more task-specific parameters does not always lead to better performance. As as shown in Tab. 7, Recon shows poor results with the vision transformer on NYUD-v2. This comparison highlights that previous multi-task expansion approaches, which naively duplicate network branches, are not only parameter-inefficient but also prone to overfitting due to the increased complexity of transformers.

Adapting to Multi-Task Architectures. In Tabs. 4 and 5, we compare DTME-MTL with leading multi-task architectures on the NYUD-v2 and PASCAL-Context datasets. We evaluate its multi-task performance against transformer-based approaches. Our method is compatible with any transformer-based multi-task architecture, enabling us to assess its effectiveness by integrating it into two leading models: InvPT and TaskPrompter. DTME-MTL seamlessly enhances these architectures, significantly boosting performance with only a minimal increase in parameters — just 0.048% for InvPT and 0.046% for TaskPrompter.

Table 6. Time consumption of each process in DTME-MTL across different backbone sizes, recorded in minutes.

Process (min)	ViT-T	ViT-S	ViT-B	ViT-L
Calculate Token Space (SVD)	3.61	3.74	11.54	11.96
Calculate Gradient Conflict	8.25	16.43	21.94	58.66

Table 7. Comparison with Recon on NYUD-v2

Method	Semseg mIoU ↑	Depth RMSE ↓	Normal mErr ↓	Edge odsF ↑	#Param ↑ (%)
Joint	34.13	0.673	22.51	56.38	0.0
Recon [22]	31.92	0.693	23.35	52.80	23.34
Ours	**38.27**	**0.6370**	**21.64**	**57.90**	**0.24**

6. Conclusion

We introduced Dynamic Token Modulation and Expansion for Multi-Task Learning, an efficient approach for improving transformer-based MTL architectures. By categorizing gradient conflicts into range space and null space, it adaptively applies token modulation and expansion to mitigate negative transfer and reduce overfitting. DTME-MTL seamlessly integrates with existing transformer-based MTL frameworks, requiring minimal additional parameters. By refining encoded token space, it provides a lightweight and scalable solution for enhancing multi-task performance.

Acknowledgment

This research was supported by the Challengeable Future Defense Technology Research and Development Program through the Agency For Defense Development(ADD) funded by the Defense Acquisition Program Administration(DAPA) in 2025(No.915102201).

References

[1] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: an end-to-end multitask learning transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12031–12041, 2022. 2

[2] Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and Jorge Cardoso. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1385–1394, 2019. 2

[3] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. *arXiv preprint arXiv:2008.10292*, 2020. 2

[4] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. *arXiv preprint arXiv:2008.10292*, 2020. 12

[5] Ronghong Cai and Jianping Luo. Multi-task learning for multi-objective evolutionary neural architecture search. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1680–1687. IEEE, 2021. 2

[6] Rich Caruana. Multitask learning. *Machine learning*, 28: 41–75, 1997. 1

[7] 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*, pages 794–803. PMLR, 2018. 3

[8] 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. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020. 3, 6, 8

[9] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11828–11837, 2023. 1, 2

[10] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 1

[11] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3150–3158, 2016. 1, 12

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 12

[13] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 2, 3

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 12

[15] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12387–12396, 2019. 1

[16] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 1, 12

[17] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep*, 2007:1–45, 2012. 12

[18] Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Advances in Neural Information Processing Systems*, 35:28441–28457, 2022. 1, 2

[19] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 12

[20] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3205–3214, 2019. 12

[21] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11543–11552, 2020. 2

[22] SHI Guangyuan, Qimai Li, Wenlong Zhang, Jiaxin Chen, and Xiao-Ming Wu. Recon: Reducing conflicting gradients from the root for multi-task learning. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3, 6, 7, 8

[23] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 270–287, 2018. 2

[24] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International conference on machine learning*, pages 3854–3863. PMLR, 2020. 2

[25] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International Conference on Machine Learning*, pages 3854–3863. PMLR, 2020. 12

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 15

[27] Huimin Huang, Yawen Huang, Lanfen Lin, Ruofeng Tong, Yen-Wei Chen, Hao Zheng, Yuexiang Li, and Yefeng Zheng. Going beyond multi-task dense prediction with synergy embedding models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28181–28190, 2024. 2

[28] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. *arXiv preprint arXiv:2103.02631*, 2021. 3

[29] Wooseong Jeong and Kuk-Jin Yoon. Quantifying task priority for multi-task optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 363–372, 2024. 3

[30] Wooseong Jeong and Kuk-Jin Yoon. Selective task group updates for multi-task optimization. *arXiv preprint arXiv:2502.11986*, 2025. 3

[31] Junguang Jiang, Baixu Chen, Junwei Pan, Ximei Wang, Dapeng Liu, Jie Jiang, and Mingsheng Long. Forkmerge: Mitigating negative transfer in auxiliary-task learning. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[32] Ian T Jollife and Jorge Cadima. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci*, 374(2065):20150202, 2016. 4

[33] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 2, 6, 8

[34] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the genetic and evolutionary computation conference*, pages 466–473, 2018. 2

[35] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021. 2, 3, 6, 8

[36] Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6, 8

[37] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. iclr, 2021. 2, 3, 6, 8

[38] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 2, 6, 8

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[40] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. 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*, pages 5334–5343, 2017. 12

[41] 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. 1, 12

[42] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1851–1860, 2019. 6, 12

[43] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 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*, pages 891–898, 2014. 6

[44] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022. 1, 2

[45] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022. 2, 3, 6, 8

[46] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. 5, 15

[47] Ramakanth Pasunuru and Mohit Bansal. Continual and multi-task architecture search. *arXiv preprint arXiv:1906.05226*, 2019. 2

[48] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 1, 2

[49] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 2, 3, 6, 8

[50] Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent component alignment for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20083–20093, 2023. 2, 3, 6, 8

[51] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 6, 12

[52] Gianluigi Silvestri. One-shot neural architecture search for deep multi-task learning in computer vision, 2020. 2

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 12

[54] Ayan Sinha, Zhao Chen, Vijay Badrinarayanan, and Andrew Rabinovich. Gradient adversarial training of neural networks. 2018. 12

[55] Guolei Sun, Thomas Probst, Danda Pani Paudel, Nikola Popović, Menelaos Kanakis, Jagruti Patel, Dengxin Dai, and Luc Van Gool. Task switching network for multi-task learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8291–8300, 2021. 12

[56] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv preprint arXiv:1904.02920*, 2019. 12

[57] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. 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*, pages 527–543. Springer, 2020. 1, 12

[58] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 2

[59] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. *arXiv preprint arXiv:2108.00154*, 2021.

[60] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2

[61] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. 1, 12

[62] Xiaogang Xu, Hengshuang Zhao, Vibhav Vineet, Ser-Nam Lim, and Antonio Torralba. Mtformer: Multi-task learning via transformer and cross-task reasoning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 304–321. Springer, 2022. 2, 8

[63] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, and Lefei Zhang. Multi-task learning with multi-query transformer for dense prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1, 2

[64] Yangyang Xu, Yibo Yang, and Lefei Zhang. Demt: Deformable mixer transformer for multi-task learning of dense prediction. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3072–3080, 2023. 1, 2

[65] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 2

[66] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 514–530. Springer, 2022. 2, 8, 12

[67] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 8, 12

[68] 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. 2, 3, 6, 8

[69] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 6

[70] Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. Mixture of attention heads: Selecting attention heads per token. *arXiv preprint arXiv:2210.05144*, 2022. 1, 2

[71] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 94–108. Springer, 2014. 1, 12

[72] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4106–4115, 2019. 1, 12