CO-Net: Learning Multiple Point Cloud Tasks at Once with A Cohesive Network

Tao Xie[1], Ke Wang[1,*], Siyi Lu[2], Yukun Zhang[3], Kun Dai[1], Xiaoyu Li[1],
Jie Xu[1], Li Wang[4], Lijun Zhao[1,5,*], Xinyu Zhang[4,*], Ruifeng Li[1,*]

[1]Harbin Institute of Technology [2]China Coal Science and Technology Intelligent Storage Technology Co., Ltd.
[3]Beijing Institute of Technology [4]Tsinghua University [5]Harbin Institute of Technology, Zhengzhou Research Institute

{xietao1997, wangke, jeff_xu, zhaolj, lrf100}@hit.edu.cn
{20s108237, 19S108222, 22S108236}@stu.hit.edu.cn 3220220839@bit.edu.cn
{wangli_thu, xyzhang}@tsinghua.edu.cn

Abstract

We present CO-Net, a cohesive framework that optimizes multiple point cloud tasks collectively across heterogeneous dataset domains. CO-Net maintains the characteristics of high storage efficiency since models with the preponderance of shared parameters can be assembled into a single model. Specifically, we leverage residual MLP (Res-MLP) block for effective feature extraction and scale it gracefully along the depth and width of the network to meet the demands of different tasks. Based on the block, we propose a novel ***nested layer-wise processing policy****, which identifies the optimal architecture for each task while provides partial sharing parameters and partial non-sharing parameters inside each layer of the block. Such policy tackles the inherent challenges of multi-task learning on point cloud, e.g., diverse model topologies resulting from task skew and conflicting gradients induced by heterogeneous dataset domains. Finally, we propose a* ***sign-based gradient surgery*** *to promote the training of CO-Net, thereby emphasizing the usage of task-shared parameters and guaranteeing that each task can be thoroughly optimized. Experimental results reveal that models optimized by CO-Net jointly for all point cloud tasks maintain much fewer computation cost and overall storage cost yet outpace prior methods by a significant margin. We also demonstrate that CO-Net allows incremental learning and prevents catastrophic amnesia when adapting to a new point cloud task.*

1. Introduction

With the substantial breakthroughs in deep neural networks, modern architectures deliver significant enhancements in the realm of point cloud analysis [27, 28, 37, 42, 61], such as 3D point classification, 3D point segmentation, and 3D point detection, etc. Nonetheless, these methods are inefficient when conducting multiple tasks since they are generally designed to execute singular task. While parallel computing can alleviate this dilemma, it may introduce additional overheads, such as memory volumes and storage expenses that increase proportionately with the quantity of tasks, which is prohibitively expensive for cutting-edge devices with limited resources (e.g., mobile devices).

Multi-task learning (MTL) [3, 14, 47, 15] provides a remedy for this problem. In visual tasks, MTL methods have primarily been introduced to jointly accomplish depth estimate task, surface normal estimate task, and semantic segmentation task from a single RGB image [16, 19, 59]. An MTL model is capable of delivering advantages in terms of complexity, inference time, and learning efficiency due to the fact that a major portion for the network can be shared across all tasks. Nonetheless, training multiple tasks concurrently for point cloud presents two critical obstacles:

i) As opposed to typical visual tasks, in which a backbone that executes admirably for image classification task can be effortlessly fine-tuned to other vision tasks, using the same backbone to jointly optimize all point cloud tasks may result in suboptimal solution for some tasks. Thus, it is preferable to find an optimal backbone for each point cloud task under resource constraint.

ii) We endeavour to operate multiple point cloud tasks concurrently by taking heterogeneous dataset domains as input rather than a regular multi-task dataset. Consequently, the gradients of different tasks will arise substantial discrepancies in the directions under multi-task learning settings, a phenomenon known as negative transfer [35].

To tackle the first challenge, we leverage residual MLP (Res-MLP) block, a basic point feature extraction block that can accommodate the requirements of various tasks in terms of the depth and width of the model architecture. Based on Res-MLP, inspired by slimmable neural networks [55, 54, 56, 4], we introduce a novel nested layer-wise pro-

*Corresponding author.

cessing policy that progressively handles the weight of each layer of the network, that is, using NAS technique to find the optimal architecture for all tasks in terms of model structure and offering fine-grained parameter sharing adaptively within the model. Compared with recent Poly-PC [53] that enables various tasks to share their common parts for a certain layer while sacrificing the flexibility of sharing, the central idea of our proposed nested layer-wise processing policy is that: (1) entangling the weights of multiple tasks within the same layer, empowering us to find optimal architectures for all tasks and achieve parameter sharing inside each layer, (2) transforming the parameter sharing of the backbone into a learnable problem so that deciding which parameters of the backbone to be shared or not can be done after training.

For negative transfer, we primarily consider conflicting directions of the gradients across various tasks, which produces a more significant impact than differences in gradient magnitude, typically for point cloud tasks, as illustrated in Table 8. Specifically, due to the manner in which gradients of various tasks are added together, gradients of multiple tasks can wipe each other out once they point to opposing directions of the parameter space, resulting in a crappy update direction for a subset or all tasks. Furthermore, CO-Net is developed for jointly streamlining multiple 3D point tasks over heterogeneous dataset domains, in which the diverse dataset domains could exacerbate such conflicting. Only very recently a few of works begin to offer ways for mitigating the conflicting gradients problem, such as eliminating conflicting portions of the gradients [57] or randomly 'dropping' pieces of the gradient vector [6]. In this work, we propose a sign-based gradient surgery that homogenizes the gradient direction of the task-shared parameters by leveraging a sign-mask manner. In this way, our proposed gradient surgery emphasizes the usage of task-shared parameters and guarantees that each task can be fully trained.

Over the well-optimized CO-Net, we perform an evolutionary searching under resource constraints to identify optimal architectures for diverse 3D point tasks. Experimental results indicate that the searched CO-Net for different tasks outperform a number of baselines and can be comparable with current state-of-the-art works optimized individually for specific tasks, as illustrated in Table 1, Table 2, and Table 3. Besides, we demonstrate that CO-Net permits incremental learning and prevents catastrophic amnesia when adapting to a new point cloud task, as shown in Table 10. Hence, CO-Net is designed to be parameter-efficient and can scale more smoothly as task numbers grow.

To summarize, the contributions of this work are as follows: 1) We propose CO-Net, a unified framework that optimizes multiple point cloud tasks collectively under various dataset domains. 2) We propose a nested layer-wise processing policy that employs NAS technique to identify the optimal architecture of different tasks while automatically determining, rather than manually, whether the parameters of the backbone are shared or not. 3) We introduce a novel sign-based gradient surgery that utilizes a sign-mask way to eliminate conflicting gradients. 4) We demonstrate that once the training for CO-Net is done, CO-Net allows incremental learning with fewer task-specific parameters by freezing the task-shared parameters.

2. Related work

Point-based methods. Point-based networks have exhibited extraordinary promise for 3D point cloud applications, i. e., point classification, detection, and segmentation. As a pioneer, PointNet [27] leverages point-wise MLP and a symmetric function to process the irregular point cloud independently. To better encode locality, PointNet++ [28] introduces the set abstraction (SA) layer to aggregate features from the points' neighborhood and proposes a hierarchical architecture based on the SA layer to learn multi-level representations of point cloud. Owing to the local point representation and multi-scale information, PointNet++ exemplifies impressive results and establishes the groundwork of modern point cloud methods [20, 46, 41, 52, 22, 31, 42, 58, 40, 45], which primarily employs graphs, convolutions, or self-attention mechanisms to perform comprehensive point cloud analysis. However, these works typically design a network to execute a single task, resulting in a linear increase in total model size with the number of tasks.

Negative transfer for multi-task learning (MTL). Compared to the typical single network training, MTL could culminate in substantial discrepancies in the directions and magnitudes of various task gradients due to skewed rivalry throughout tasks for the shared parameters. Previous researches [5, 14, 18, 23, 36, 38] propose substantial algorithms to homogenize these differences. For gradient magnitudes, GradNorm [5] leverages loss descent to represent learning speed, which are dynamically adjusted to preserve the same learning pace of each task. RotoGrad [14] homogenizes the gradient magnitudes through normalizing and scaling, ensuring training convergence. For gradient directions, current researches [9, 57, 34] typically investigate the cosine similarity between gradients to alleviate the arduous training with multiple objectives. The cosine similarity is always considered as a regularization or an indicator. [34] adds a regularization factor that compels the cosine similarity between two distinct losses to be greater than zero. In [9], once the cosine similarity is negative (gradient conflict), the auxiliary task's weight is set to zero. [57] projects the gradient of each loss to achieve orthogonal gradients for all tasks. In this work, we propose a novel sign-based gradient surgery that eliminates conflicting gradient directions in a sign-mask manner.

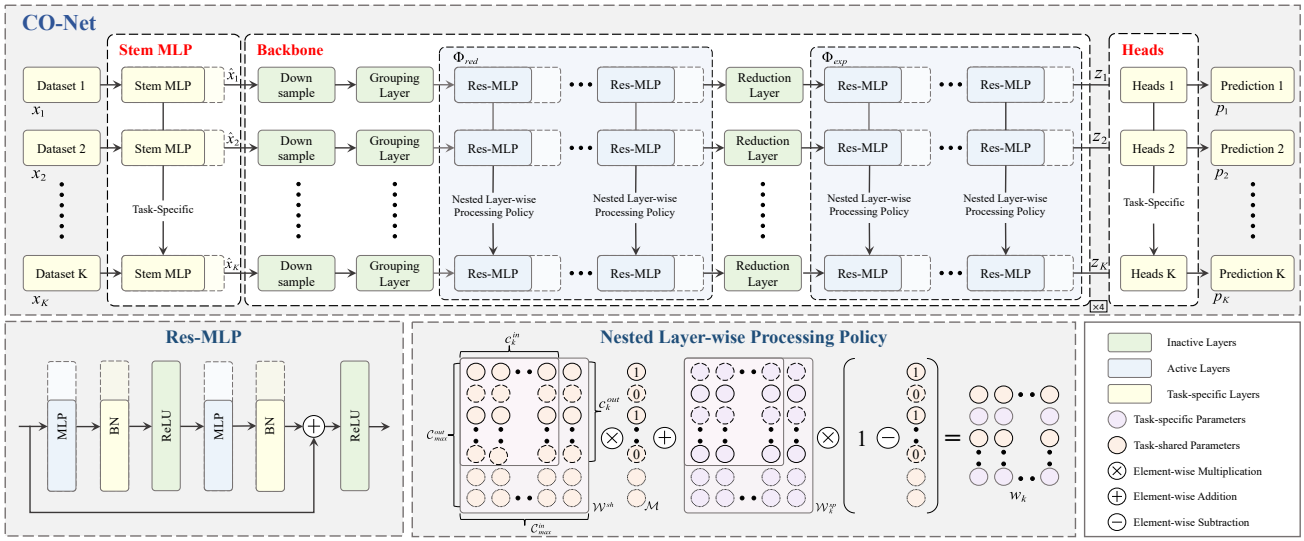

Figure 1. **The network architecture of CO-Net.** CO-Net jointly optimizes multiple point cloud tasks and consists of three components, that is, stem MLP, backbone, and heads for various tasks. The nested layer-wise processing policy is applied to each layer of the backbone to identify optimal architectures for different tasks and automatically decide which parameters of the backbone to be shared, through which we can generate task-related features. The parts in solid lines mean they are chosen while those in dashed lines are not.

3. Methodology

CO-Net aims to optimize K different point cloud tasks concurrently, that is, finding K mappings from K different datasets $\{\mathcal{X}_k\}$ to a task-specific set of labels $\{\mathcal{Y}_k\}$, $k = 1, 2, ..., K$. In this work, we focus on the design and optimization of CO-Net, which encourages all tasks to share parameters as many as feasible for efficient storage whereas retaining superior performance for all tasks.

3.1. Design CO-Net

As shown in Fig. 1, CO-Net is comprised of three components: stem MLP, a unified backbone, and K heads for K tasks.

Stem MLP. As illustrated in Section 1, as CO-Net takes ross heterogeneous dataset domains as input whereas the dimensions of these dataset domains are various, we utilize a stem MLP to map the input $x_k \in \mathcal{X}_k$ into the identical dimension as $\hat{x_k}$. Each task k possesses its own individual parameters ϕ_k^{sp}.

Backbone. Following typical architecture design of current works [13, 28, 22], the backbone consists of four stages for point feature extraction, with each stage leveraging a subsampling layer to downsample the incoming points, a grouping layer to query neighbors for each point, a stack of Res-MLP blocks to extract features, a reduction layer to aggregate features within the neighbors, and a symmetrical Res-MLP block to extract deep aggregated features. Conceptually, the kernel operation of each stage in CO-Net can be formulated as:

$$g_i = \Phi_{exp}(\mathcal{R}((\Phi_{red}([f_{i,j}; (x_{i,j} - x_i)/r])))), \quad (1)$$

where r is the group radius; x_i is the i-th sampled point coordinate; $x_{i,j}$ and $f_{i,j}$ are the coordinate and feature of the j-th neighbor point of x_i respectively; Φ_{red} and Φ_{exp} are depth-alterable Res-MLP blocks in each stage, with each Res-MLP block consisting of two MLP layers, two normalization layers, and two ReLU activation layers, as illustrated in the bottom left of Fig. 1; $\mathcal{R}$ is the reduction layer (e.g. max-pooling) that aggregates features for point i from its neighbors. By integrating residual connections in the Res-MLP block, CO-Net can be easily stretched to dozen layers without vanishing gradient. In addition, bottleneck design is applied to each Res-MLP block of Φ_{red} to reduce computation, whereas inverted bottleneck design is applied to each Res-MLP block of Φ_{exp} to reinforce point-wise feature extraction. Notably, for the last Res-MLP block of Φ_{red} in each stage, we leverage the second MLP of the block to map the input into the predefined output channel, and add a 1×1 convolution with a normalization layer in the residual connection.

Overall, the backbone concludes a set of task-shared parameters θ^{sh} for K tasks and task-specific parameters θ_k^{sp} for task k to transform each input $\hat{x}_k$ into an intermediate representation $z_k = f(\hat{x}_k; \theta^{sh}, \theta_k^{sp})$.

Heads. Besides, each task k has a head network h_k, with its exclusive parameters ψ_k^{sp}, to take z_k as input and outputs the prediction $p_k = h_k(z_k; \psi_k^{sp})$ for the corresponding task.

Nextly, we discuss how to enable CO-Net to perform multiple point cloud tasks collectively. As illustrated in Section 1, the backbone for different point cloud tasks must be designed carefully. Thus, we propose a nested layer-wise processing policy that utilizes NAS technique to iden-

tify optimal architectures for various tasks and leverages a learnable score to adaptively determine whether the parameters of the backbone are shared or not. We presume the layers with learnable parameters in the backbone to be active layers, such as convolution and normalization layers, whereas other layers, such as grouping layer, activation layer and reduction layer, are considered as inactive layers.

3.2. Nested Layer-wise Processing Policy

Since each task has its own dataset domain, we distribute different tasks on different GPUs, with K tasks corresponding to K GPUs. Taking one MLP layer of the Res-MLP block as an example, we define two search spaces for the layer, i.e., input channel $\mathcal{C}_k^{in}$ and output channel $\mathcal{C}_k^{out}$ of task k. Accordingly, we initialize the weight of the layer as $\mathcal{W} \in \mathbb{R}^{\mathcal{C}_{max}^{out} \times \mathcal{C}_{max}^{in}}$, where $\mathcal{C}_{max}^{out}$ and $\mathcal{C}_{max}^{in}$ are the maximum number of $\{\mathcal{C}_1^{out}, \mathcal{C}_2^{out}, ..., \mathcal{C}_K^{out}\}$ and $\{\mathcal{C}_1^{in}, \mathcal{C}_2^{in}, ..., \mathcal{C}_K^{in}\}$ respectively. Employing a single weight for each layer, however, does not enable different tasks to share or monopolize the parameters dynamically at various iterations during training, hence restricting the adaptability of CO-Net for the tasks.

To tackle this issue, we consider the weight $\mathcal{W}$ as task-shared weight $\mathcal{W}^{sh}$ and introduce two additional parameters: task-specific weight $\mathcal{W}_k^{sp} \in \mathbb{R}^{\mathcal{C}_{max}^{out} \times \mathcal{C}_{max}^{in}}$ and learnable score $\mathcal{S}_k = [\mathcal{S}_k^1, \mathcal{S}_k^2, ..., \mathcal{S}_k^{\mathcal{C}_{max}^{out}}] \in \mathbb{R}^{\mathcal{C}_{max}^{out}}$. We define an indicator function to judge whether the parameters of the layer are shared or not in current iteration, formulated as:

$$\Theta(\mathcal{S}_k^j) = \begin{cases} 1 & if \ \ \mathcal{S}_k^j \geq \lambda \\ 0 & otherwise \end{cases}, \tag{2}$$

where λ is a threshold; if $\mathcal{S}_k^j \geq \lambda$, the j-th channel of the layer is transformed to task-shared parameters and consequently optimizing such parameters in the global group. Towards this end, we can obtain the task-sharing mask $\mathcal{M}$ as follows:

$$\mathcal{M} = [\Theta(\mathcal{S}_k^1), \Theta(\mathcal{S}_k^2), ..., \Theta(\mathcal{S}_k^j), ..., \Theta(\mathcal{S}_k^{\mathcal{C}_{max}^{out}})]. \tag{3}$$

For each batch during supernet training, we sample a number c_k^{out} from $\mathcal{C}_k^{out}$ and c_k^{in} from $\mathcal{C}_k^{in}$. Based on the sampled channel and the sharing mask $\mathcal{M}$, we slice out the task-shared weight for task k by

$$w_k^{sh} = \mathcal{W}^{sh}[: c_k^{out}, : c_k^{in}] \otimes \mathcal{M}[: c_k^{out}], \tag{4}$$

and slice out the task-specific weight for current batch by

$$w_k^{sp} = \mathcal{W}_k^{sp}[: c_k^{out}, : c_k^{in}] \otimes (1 - \mathcal{M})[: c_k^{out}], \tag{5}$$

where $\otimes$ denotes the element-wise multiplication.

Finally, we can obtain the current weight as w_k by

$$w_k = w_k^{sh} + w_k^{sp}, \tag{6}$$

where w_k is used to produce the output for current batch of the task k, as illustrated in the bottom right of Fig. 1.

During backward pass, we denote the gradient of w_k^{sh} as G_k^{sh}. To achieve parameter sharing, the gradients across all GPUs are averaged. Since gradient G_k^{sh} is not consistent, induced by non-equivalence between the search spaces $\mathcal{C}_k^{in}$ and $\mathcal{C}_k^{out}$ of different tasks, we pad the gradient with zero to the largest size of the gradient by

$$\hat{G}_k^{sh} = [G_k^{sh}, 0] = max(G_1^{sh}, G_2^{sh}, ..., G_K^{sh}). \tag{7}$$

In this way, we can average the gradients across all GPUs to update the task-shared weight. If we ensure that the initialization of task-shared weight, learning rate, and weight decay are the same on all GPUs, the task-shared weight $\mathcal{W}^{sh}$ across all GPUs would always keep same throughout training. For the task-specific weight, we directly update the parameters by using the gradient on each GPU.

We apply above procedure on all active layers of the backbone to enable different tasks to share or monopolize the parameters dynamically at various iterations. Besides, concluding additional parameters $\mathcal{W}_k^{sp}$ and S_k into each layer does not result in a large increase in memory cost since only the selected parameters w_k are optimized at each iteration and all other parameters are kept offline. Moreover, during the inference and model deployment phase, we can slice out the parameters w_k^{sh} and corresponding w_k^{sp} for all tasks, and deposit weights of all tasks into one to achieve efficient storage deployment.

It is worth noting that the learnable score $\mathcal{S}_k$ is set as task-shared to ensure the parameters of all tasks can be integrated into a compact model, that is, $\mathcal{S} = \mathcal{S}_1 = \mathcal{S}_2 = ... = \mathcal{S}_K$. Since the indicator function $\Theta(\cdot)$ is not differentiable, we need to modify its gradient during backward pass, which will be presented in Section 3.4. Moreover, as illustrated in Table 7, learning task-specific normalization can significantly improve the performance for all tasks while adding a few parameters, so we set the parameters of normalization layers as task-specific.

3.3. Sign-based Gradient Surgery

We seek to optimize architecture parameters of CO-Net by simultaneously learning multiple point cloud tasks. For convenient portrayal, we define the task-shared and task-specific parameters of CO-Net as $\mathcal{X}^{sh} = \{\theta^{sh}\}$ and $\mathcal{X}_k^{sp} = \{\phi_k^{sp}, \theta_k^{sp}, \psi_k^{sp}\}$ respectively.

For task-specific parameters $\mathcal{X}_k^{sp}$, we obtain the gradient of $\mathcal{X}_k^{sp}$ for the k-th task at the i-th iteration as:

$$G_{k,i}^{sp} = \nabla_{\mathcal{X}_k^{sp}} \mathcal{L}_{k,i}, \tag{8}$$

where $\mathcal{L}_{k,i}$ means the loss function for the k-th task at the i-th iteration. Subsequently, the task-specific parameters on each GPU will be optimized based on the calculated $G_{k,i}^{sp}$.

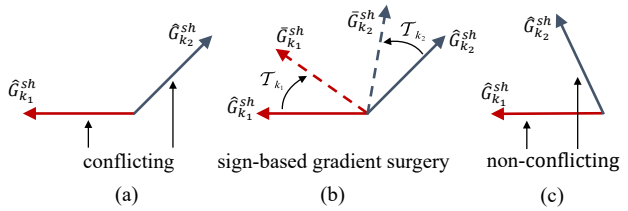

Figure 2. **The illustration of sign-mask gradient surgery.** In (a), the gradient directions of the k_1-th task and the k_2-th task are in conflict, i.e., their cosine similarity is negative, resulting in destructive interference. In (b), we construct consensus gradient $\bar{G}^{sh}_{k_1,i}$ and $\bar{G}^{sh}_{k_2,i}$ in the case where $\hat{G}^{sh}_{k_1,i}$ and $\hat{G}^{sh}_{k_2,i}$ are in conflict. In (c), non-conflicting task gradients are not altered by our algorithm, allowing for positive parameter optimization.

Noting that in the context of optimizing a task with multiple GPUs, the gradients $G^{sp}_{k,i}$ on these GPUs would be averaged and then the parameters are updated accordingly.

For task-shared parameters $\mathcal{X}^{sh}_k$, the gradient of $\mathcal{X}^{sh}_k$ for the k-th task at the i-th iteration can be also formulated as

$$\hat{G}^{sh}_{k,i} = \nabla_{\mathcal{X}^{sh}_k} \mathcal{L}_{k,i}. \tag{9}$$

To tackle the issue that the task-shared parameters of K tasks have different gradient sizes, we employ Eq. (7) to obtain the processed gradients $\hat{G}^{sh}_{k,i}$ with the same dimension.

As outlined in Section 3.2, a straightforward scheme to optimize the task-shared parameters involves averaging the gradients $\hat{G}^{sh}_{k,i}$ across all GPUs and then updating the corresponding weights. Although such scheme simplifies the optimization problem, it may also spark gradient conflict among tasks, leading to an overall performance drop due to a skewed competition among tasks for the shared parameters. Additionally, CO-Net is designed to optimize multiple point cloud tasks simultaneously, spanning different dataset domains with varying data distributions. In this way, negative transfer will be further amplified since the optimization directions may diverge under the conditions of multiple data distributions. Inspired by [9, 57, 34, 14], we design a sign-based gradient surgery that homogenizes the gradient directions of the task-shared parameters for all tasks by leveraging a sign-mask way, thus emphasizing the usage of task-shared parameters.

We employ the gradient inner product to appraise the consistency of gradient directions. If the gradient inner product is negative, the gradient directions of any two tasks are deemed conflicting, as illustrated in Fig. 2 (a) (conflicting) and Fig. 2 (c) (non-conflicting). We design a sign-mask method to reduce the effect of conflicting gradients. We first introduce the definition of positive gradient possibility, which is formulated as:

$$\mathcal{P}[m] = \frac{1}{2} + \frac{1}{2} \cdot \frac{\sum_k \hat{G}^{sh}_{k,i}[m]}{\sum_k \left|\hat{G}^{sh}_{k,i}[m]\right|}, \; 0 \le \mathcal{P}[m] \le 1, \tag{10}$$

where $\mathcal{P}[m]$ represents the positive value possibility of the m-th component among K gradients. Then, we compute the m-th component of $\mathcal{T}_k$ for the k-th task according to $\mathcal{P}[m]$, formulated as:

$$\mathcal{T}_k[m] = \begin{cases} 1 & sgn(\hat{G}^{sh}_{k,i}[m]) > 0 \;\; and \;\; \mathcal{P}[m] > \bar{\mathcal{S}} \\ 1 & sgn(\hat{G}^{sh}_{k,i}[m]) < 0 \;\; and \;\; 1 - \mathcal{P}[m] > \bar{\mathcal{S}} \\ 0 & otherwise \end{cases}, \tag{11}$$

where $\mathcal{T}_k[m]$ is the sign-mask of the m-th component for the k-th task, $sgn(\cdot)$ is the signal function, $\bar{\mathcal{S}}$=$Sigmoid(\mathcal{S}_{avg})$ in which $\mathcal{S}_{avg}$ denotes the average value of the learnable scores throughout the backbone. Notably, we employ $\bar{\mathcal{S}}$ to adaptively identify the sign-mask $\mathcal{T}$ at each iteration, with the key insight that: the value of $\bar{\mathcal{S}}$ can implicitly reflect the sharing ratio of parameters at each iteration, and theoretically the more shared parameters, the greater probability of gradient conflict, which implies utilizing a larger threshold for the computation of $\mathcal{T}$. In this way, we select $\bar{\mathcal{S}}$ as the threshold is appropriate.

The next step is to determine the value of each component for the consensus gradient $\bar{G}^{sh}_{k,i}$ of task k. The value of the m-th component is defined as follows:

$$\bar{G}^{sh}_{k,i}[m] = \hat{G}^{sh}_{k,i}[m] \cdot \mathcal{T}_k[m], \;\; m = 1, 2, ..., M, \tag{12}$$

where M is the total size of the gradient vectors $\hat{G}^{sh}_{k,i}$, that is given by the number of task-shared parameters, i.e., $M = \left|\mathcal{X}^{sh}\right|$.

Finally, the gradient vector G^{sh}_i of the shared parameters is formulated as:

$$G^{sh}_i = \frac{1}{K}(\bar{G}^{sh}_{1,i} + \bar{G}^{sh}_{2,i} + ... + \bar{G}^{sh}_{k,i} + ... + \bar{G}^{sh}_{K,i}), \tag{13}$$

where G^{sh}_i is used to update the shared parameters $\mathcal{X}^{sh}$.

Compared to previous gradient homogenization algorithms, our proposed sign-based gradient surgery uses a sign-mask strategy to zero out conflicting gradients of one task with other tasks, as illustrated in Fig. 2 (b).

3.4. Gradient Estimation of $S_{k,i}$

In the backward pass of CO-Net, the gradient of the score $S_{k,i}$ for the k-th task at the i-th iteration is formulated as:

$$\nabla_{S_{k,i}} \mathcal{L}_{k,i} = \frac{\partial \mathcal{L}_{k,i}}{\partial \Theta(S_{k,i})} \frac{\partial \Theta(S_{k,i})}{S_{k,i}}. \tag{14}$$

Direct application of gradient descent for optimization is not feasible since the gradient of the indicator function $\Theta(\cdot)$ is zero at almost all points. We choose a straight gradient estimator to tackle this issue and the modified gradient of $S_{k,i}$ is given by $\nabla_{S_{k,i}} \mathcal{L}_{k,i} = \frac{\partial \mathcal{L}_{k,i}}{\partial \Theta(S_{k,i})}$.

3.5. Search pipeline

Training CO-Net. At each training iteration, we uniformly sample a subnet from search space (detailed in Appendix A.1) for each task, with the corresponding weights in CO-Net being updated while the rest is frozen. Notably, the task-shared parameters are optimized by the proposed sign-based gradient surgery whereas the task-specific parameters are optimized by Eq. (8). More details are given in Appendix A.2.

Searching CO-Net. The optimum subnet for each task will be acquired through evolutionary search over the well-trained CO-Net. In the searching process, we evaluate and select the subnets in accordance with the manager of the evolution algorithm [4] to accomplish the maximization of the proxy performance and the minimization of the model size (model parameters) over each task. A detailed process is also given in Appendix A.3.

4. Experiment

In this part, we first evaluate the performance of CO-Net when jointly optimizing three tasks, including 3D point classification, 3D semantic segmentation, and 3D object detection. Then we validate the inner design of CO-Net and its incremental learning ability when adapted to a new task.

4.1. Datasets and Metric

ScanObjectNN [43]. The ScanObjectNN is commonly applied in 3D object classification, comprising 15 classes with 2,902 unique object instances scanned from about 15,000 real objects. ScanObjectNN delivers significant obstacles to conventional point cloud analysis techniques due to its obstructions and noise. Following PointMLP [22], we execute experiments on PB_T50_RS, the most challenging and prevalent ScanObjectNN variant. We leverage the overall accuracy (OA) across all classes as evaluation metric.

S3DIS [1]. The S3DIS dataset consists of 271 rooms across three buildings, which is divided into six sections. We utilize the typical overall point-wise accuracy (OA) and mean class-wise intersection over union (mIoU) to assess CO-Net on this dataset. We withhold area 5 when training and employ it for testing to evaluate CO-Net.

SUN RGBD [39]. SUN RGBD dataset is comprised of around 5,000 RGB-D training images tagged with 3D bounding boxes for 37 item classes. We adhere to the standard data split [26, 21] and data processing protocols of VoteNet [26] for the dataset. What is more, the results of SUN RGBD with mean average precision (mAP) at different IoU thresholds, 0.25 and 0.5 are released.

4.2. Implementation Details

Search space. We construct an expansive search space that incorporates multiple variable elements in each stage of the backbone: the number of neighbour points, group radius, the number of Res-MLP blocks in Φ_{red} and Φ_{exp}, reduction rate in Φ_{red}, expansion rate in Φ_{exp}, and output channels of current stage. The output channel of the stem MLP is also designed to be searchable. Notably, each point cloud task possesses its individual searching space. Appendix A.1 provides an extensive overview of the search space for CO-Net (base) and CO-Net (large).

Training CO-Net. CO-Net is an end-to-end pipeline optimized by the initial learning rate 0.008 with cosine annealing and AdamW optimizer with a weight decay 0.05. The batch size is set to 16 for ScanObjetNN, 8 for S3DIS and SUN RGBD. We allocate 4 Tesla A100 GPUs for the training of classification, segmentation and detection respectively, with a total training of 100,000 iterations for each task. The same head with PointNet++ [28] is utilized for 3D point classification and 3D semantic segmentation while the identical head with VoteNet [26] is adopted for 3D object detection. We embed the upsampling layer within the head of each task for feature propagation [28], the parameters of which are regarded as task-specific. The data augmentations offered in [30] are employed for 3D point classification and semantic segmentation. Data augmentations provided in VoteNet [26] are manipulated for 3D object detection. For all of our experiments, we set λ in Eq. (2) as 0.5.

Searching CO-Net. Following the same protocol as in [11, 4, 53], we implement the evolution search algorithm with a population size of 50 and a generation number of 20. At each generation, we choose the top 10 architectures to serve as progenitors for the generation of offspring networks via mutation and crossover.

4.3. Comparison with State-of-the-art Methods

In this part, we evaluate CO-Net in comparison with baselines and current state-of-the-art methods on ScanObjectNN dataset, S3DIS dataset, and SUN RGBD dataset, with the results reported in Table 1, Table 2, and Table 3.

Method	Points	Flops (G)	#Params (M)	OA
PointCNN [17]	1k	1.6	0.6	78.5
DGCNN [48]	1k	4.8	1.8	78.1
GBNet [32]	1k	-	8.8	80.5
SimpleView [10]	1k	-	0.8	80.5
PRANet [8]	1k	-	1.2	82.1
MVTN [12]	1k	1.8	3.5	82.8
PointMLP [22]	1k	31.3	13.2	85.4
PointNeXt [30]	1k	1.4	1.6	87.7
RepSurf-U [33]	1k	0.8	1.5	84.6
PointNet++ [28]	1k	1.7	1.5	77.9
PointNet++† [28]	1k	1.7	1.5	86.1
CO-Net (base)	1k	2.8	1.7	88.4 (+10.5)
CO-Net (large)	1k	4.1	5.7	89.5 (+11.6)

Table 1. The results of 3D point classification on ScanObjectNN dataset. #Params denotes the number of parameters. † denotes the results reported in PointNeXt [30].

ScanObjectNN. As shown in Table 1, we implement the

experiments with 1k points as input on ScanObjectNN and the results demonstrate the superority of CO-Net. Specifically, CO-Net (base) outpreforms PointNet++ [28] and PointNet++† by 10.5 units and 2.3 units in terms of overall accuracy with the increase of 1.1G FLOPs and 0.2M number of parameters. Furthermore, the large model CO-Net (large) improves the overall accuracy to 89.5 units, exceeding current state-of-the-art methods PointNeXt [30] and PointMLP [22] by 1.8 units and 4.1 units with comparable FLOPs and parameters.

Method	Flops (G)	#Params (M)	OA	mIoU
PointCNN [17]	-	0.6	85.9	57.3
PCCN [2]	7.3	5.4	-	58.3
Kpconv [42]	2.1	14.9	-	67.1
ASSANet [29]	2.5	2.4	-	63.0
ASSANet-L [29]	36.2	115	-	66.8
Point Trans. [61]	7.8	5.6	90.8	70.4
PointNeXT-S [30]	3.6	0.8	87.9	63.4
PointNeXT-B [30]	8.8	3.8	89.4	67.5
RepSurf-U [33]	1.0	1.0	90.2	68.9
Poly-PC (base) [53]	1.0	1.0	88.2	63.0
Poly-PC (large) [53]	5.6	5.6	89.5	66.0
PointNet++* [28]	1.0	1.0	85.8	56.9
PointNet++† [28]	1.0	1.0	87.5	63.2
CO-Net (base)	1.2	1.5	89.7 (+3.9)	65.4 (+8.5)
CO-Net (large)	10.6	2.7	90.4 (+4.6)	68.0 (+11.1)

Table 2. 3D point semantic segmentation results on the S3DIS dataset, evaluated on area 5.

S3DIS. Table 2 illustrates the comparison results of CO-Net with other state-of-the-art methods under the S3DIS dataset. CO-Net (base) achieves 65.4 mIoU and 89.7 overall accuracy respectively, which exceeds PointNet++ [28] by 8.5 units and 3.9 units with an acceptable increase of FLOPs and parameters. Again, CO-Net (base) demonstrates its preeminence over PointNeXT-S [30], exceeding it by 2.0 units in mIoU and 1.8 units in overall accuracy. Compared to the state-of-the-art methods KPConv[42], ASSANet-L[29] and PointNext-B[30], CO-Net (large) also achieves competitive results in the condition of fewer FLOPs and parameters. Compared with Poly-PC [53], CO-Net families surpass it by large margins, revealing that the significance of offering fine-grained parameter sharing adaptively within the model. It seems that CO-Net may not achieve the same level of performance as current state-of-the-art segmentation methods (e.g., Point Transformer [61] and RepSurf [33]) deliberately designed for segmentation task, however, the utilization of a framework to simultaneously accomplish multiple point cloud tasks is figured out through the experiment, which is conducive to final model deployment.

SUN RGBD. We evaluate CO-Net under SUN RGBD dataset against several competitive approaches and the results are exhibited in Table 3. CO-Net (base) possesses an outstanding performance compared with the baseline VoteNet/VoteNet* [26], 3.5/1.2 units improvement under

*We report the results of MMDetection3D (https://github.com/openmmlab/mmdetection3d), which are superior to those of the official paper.

Method	Flops (G)	#Params (M)	mAP@0.25	mAP@0.5
ImVoteNet* [25]‡	241	42.5	64.5	-
3Detr [24]	9.8	7.0	59.1	32.7
MLCVNet [51]	7.2	1.2	59.8	-
H3DNet [60]	14.5	6.3	60.1	39.0
BRNet [7]	8.0	3.2	61.1	43.7
VENet [50]	30.3	4.9	62.5	39.2
Group-free* [21]	11.2	19.8	63.0	45.2
RBGNet [44]	3.5	2.2	64.1	47.2
RepSurf-U [33]	-	11.5	64.3	45.9
Poly-PC (base) [53]	6.1	1.0	62.3	40.2
Poly-PC (large) [53]	18.8	7.8	63.5	41.9
Farp-Net [52]	-	1.8	64.0	-
VoteNet* [26]	5.8	1.0	59.1	35.8
VoteNet* [26]	5.8	1.0	61.4	37.9
CO-Net (base)	6.6	1.1	62.6 (+3.5)	41.1 (+5.3)
CO-Net (large)	16.4	6.6	63.7 (+4.6)	44.6 (+8.8)

Table 3. The performance of CO-Net against previous works on 3D object detection under SUN-RGBD. Note that ‡ means it uses RGB as extra inputs whereas CO-Net is geometric only. * means our implementation, detailed in Appendix F.3.

Method	OA	mIoU	mAP@0.25	S-Score	#Params-t (M)
Baseline	77.9	56.9	59.1	193.9	3.500
RepSurf-U [33]	84.6	68.9	64.3	217.8	13.959
CO-Net (base)	88.4	65.4	62.6	216.1 (+22.2)	3.362 (-0.138)
CO-Net (large)	89.5	68.0	63.7	221.2 (+27.3)	10.431 (+6.931)
PointNet++† [28]	86.1	63.2	-	149.3	2.500
PointCNN [17]	78.5	57.3	-	135.8	1.200
PointNeXT-S [30]	87.7	63.4	-	151.1	2.400
PointNeXT-B [30]	87.7	67.5	-	155.2	5.400
CO-Net* (base)	88.4	65.4	-	153.8	2.731
CO-Net* (large)	89.5	68.0	-	157.5	6.116

Table 4. The overall performance of CO-Net with baseline and other state-of-the-art methods. #Params-t denotes the total parameters when jointly optimizing multiple tasks.

mAP@0.25 and 5.3/3.2 units under mAP@0.5 with few parameters increasing. Additionally, CO-Net (large) reaches 63.7 on mAP@0.25 and 44.6 mAP@0.5, achieving competitive results with current state-of-the-arts Groupfree [21], RBGNet [44], Poly-PC [53], Farp-Net [52], and RepSurf [33]. Despite it turns out to operate without intricate heads like Groupfree [21] and RBGNet [44], CO-Net still acquires comparable performance with them and surpasses many previous works [50, 7] that only designed for detection task, indicating the significance of a strong backbone for boosting performance.

Overall performance. To further highlight the superiority of CO-Net, we compare the overall performance of CO-Net on the above three tasks with that of the baselines (e.g., PointNet++ [28] and VoteNet [26]), as well as with the total model parameters for concurrently optimizing these tasks. In this process, PointNet++ [28] and VoteNet [26] are incorporated as the baselines as CO-Net utilize the identical heads with them. In addition, we define CO-Net* for 3D point classification and 3D semantic segmentation tasks such that comparing it with current cutting-edge works that typically evaluate performances on these two tasks. We designate the S-Score as a proxy for overall performance acquired by the linear summation of perfor-

	Training scopes	cls	seg	det	#Params-t (M)
CO-Net (base)	individual	88.6	64.8	62.9	4.300
CO-Net (base)	joint&nas	88.4	65.4	62.6	3.362
CO-Net (large)	individual	89.9	67.6	63.8	15.000
CO-Net (large)	joint&nas	89.5	68.0	63.7	10.431

Table 5. The overall performance of CO-Net under joint training scope and individual training scope.

	Training scopes	cls	seg	det	#Params-t (M)
random	individual	87.3	63.6	62.1	4.751
CO-Net (base)	individual	88.6	64.8	62.9	4.300
random	individual	88.4	62.7	62.6	15.762
CO-Net (large)	individual	89.9	67.6	63.8	15.000

Table 6. The effect of NAS technique in the nested layer-wise processing policy.

	cls	seg	det	S-Score	#Params (M)
EXP1	85.8	63.2	61.1	210.1	2.698
EXP2	87.6	64.5	62.3	214.4	3.306
EXP3	88.4	65.4	62.6	216.1	3.362

Table 7. The effect of different parameter sharing strategies.

SGS	PCGrad	DWA	cls	seg	det	S-Score
✗	✗	✗	87.0	64.3	61.9	213.2
✓	✗	✗	88.4	65.4	62.6	216.1
✗	✓	✗	87.9	65.1	62.1	215.1
✗	✗	✓	87.3	64.8	61.9	214.2

Table 8. Results of CO-Net (base) with various gradient homogenization algorithms to resolve negative transfer. SGS denotes the proposed sign-based gradient surgery.

λ	cls	seg	det	S-Score
0.25	88.1	65.2	62.4	215.4
0.50	88.4	65.4	62.6	216.1
0.75	88.6	64.9	62.3	215.5

Table 9. The ablation results for λ.

mances in all tasks. The results in Table 4 demonstrate that CO-Net (base) exceeds the baseline by 22.2 units in the S-Score, accompanied by a decrease of 0.138M parameters. CO-Net (large) outperforms RepSurf-U [33] by 3.4 units in S-Score with 3.528M fewer #Params-t, highlighting the superiority of CO-Net. CO-Net* (base/large) secures comparable results when compared with the current cutting-edge methods, which merely evaluate performances on 3D point classification and 3D semantic segmentation tasks. Specifically, CO-Net* (base) achieves superior performance relative to PointNet++† [28]/PointNeXT-S [30], with 4.5/2.7 units increase in terms of S-Score with comparable parameters. CO-Net* (large) also exceeds current state-of-the-art PointNeXT-B [30] by 2.3 units regarding to S-Score with 0.716M parameters increase.

4.4. Ablation Study

The effect of multi-task training for point cloud. We retrain the optimal model architecture searched by CO-Net for each point cloud task from scratch and compare the results with CO-Net for all point cloud tasks, to verify the effectiveness of CO-Net. As demonstrated in Table 5, the performance of subnets produced from CO-Net is comparable to that trained separately, with somewhat lower accuracy in classification and detection tasks while higher in segmentation task. Notably, CO-Net (base)/CO-Net (large) acquire a storage margin of 0.938M/4.569M, respectively, due to the fact that the majority of parameters in CO-Net are shared across all tasks.

The effect of nested layer-wise processing policy. As illustrated in Section 3.2, our proposed nested layer-wise processing policy leverages NAS technique to identify the optimal architecture for each task while automatically determining whether the parameters of backbone are shared or not. Thereby, we evaluate the effectiveness of the NAS technique to determine the optimal architecture for various tasks. Particularly, we randomly select three networks under parameter constraint and train them independently from scratch for different tasks. The results compared with networks retrieved by CO-Net are summarized in Table 6. CO-Net substantially achieves almost 1 unit increase on all three tasks, whilst the parameters yield only a minor change, illustrating the superiority of the NAS technique in the nested layer-wise processing policy. Subsequently, to present the potency of sharing policy in nested layer-wise processing policy, we apply three distinct parameter sharing strategies to CO-Net (base). EXP1: All parameters of the backbone are set as task-shared. EXP2: All parameters of the layers (both convolutional and normalization layers) in the backbone are determined whether to be shared by scores. EXP3: All parameters of the layers (only convolutional layers) in the backbone are determined whether to be shared by scores. As demonstrated in Table 7, compared to set all parameters as task-shared, CO-Net significantly improves the overall performance of all tasks from 210.1 to 216.1 at the expense of a small increase in parameters, which indicates that utilizing additional task-specific parameters to learn task-related features is essential for effective feature extraction. Besides, the performance of CO-Net is further enhanced with normalization layers set as task-specific, i.e., EXP2 vs EXP3.

The effect of sign-based gradient surgery. In this section, the results of CO-Net (base) with and without sign-based gradient surgery are first illustrated in Table 8. In addition, we train the network along with other typical algorithms (e.g., PCGrad [57] and Dynamic Weight Average [19]) designed for tackling negative transfer in multi-task learning tasks. Our method achieves superior performance in comparison to PCGrad [57] and DWA [19], further demonstrating the effectiveness of the proposed sign-based gradient surgery. Particularly, the overall performance of our algorithm surpasses that of DWA by 1.9 units, indicating that homogenizing the conflicting directions of the gradients for different tasks is more significant.

The selection of λ. In this part, we conduct an ablation

Method	Points	FLOPs (G)	#Params-e	OA
PointNet++ [28]	1k	3.2	100%	90.7
PointNet++ [28]	5k	3.2	100%	91.9
CO-Net (base)	1k	2.9	62.15%	92.1
CO-Net (large)	1k	5.9	58.11%	92.9

Table 10. Incremental learning. #Params-e indicates that the percentage of extra parameters required by each method when generalizing to a new task.

experiment to verify the selection of λ, with the results reported in Table 9. When setting λ to 0.5, CO-Net achieves the best performance compared with other settings.

4.5. Incremental learning of CO-Net

When CO-Net is applied to a new task, only the task-specific parameters require to be retrained, while the task-shared parameters will be frozen due to their capacity to encapsulate generalized features of the point cloud. It is concluded that CO-Net enables incremental learning and can scale smoothly with the task numbers increasing. To exemplify this concept, we implement the Human-made Object Classification ModelNet40 [49] into CO-Net and compare it to PointNet++ [28]. The training recipe and details of ModelNet40 are given in Appendix B. Notably, we use the searched architecture for ScanObjectNN to conduct the incremental experiment on ModelNet40 dataset. The incremental results of CO-Net are summarized in Table 10. We observe that CO-Net (base/large) achieves 92.1/92.9 overall performance with fewer parameter proportions, further indicating that CO-Net is capable of achieving genuinely incremental learning. The catastrophic forgetting analysis is also detailed in Appendix B.

4.6. Generalization Of CO-Net

To evaluate the flexibility of CO-Net, following Poly-PC [53], we utilize CO-Net to jointly optimize ModelNet40, S3DIS, and SUN RGBD, and then conduct incremental experiments (IE) on ScanObjectNN, with results summarized in Table 11. Under such setting, CO-Net (base/large) exceed Poly-PC (base/large) [53] by non-trivial margin in terms of S-Score while maintain much fewer total parameters, emphasizing the significance of offering fine-grained parameter sharing adaptively within the model.

Method	OA	mIoU	mAP@0.25	S-Score	#Params-t (M)	OA (IE)
Poly-PC (base) [53]	92.6	63.0	62.3	217.9	3.4	86.8
Poly-PC (large) [53]	93.7	66.0	63.5	223.2	13.7	87.9
CO-Net (base)	93.2	66.1	62.5	221.8	3.314	87.1
CO-Net (large)	94.1	68.4	63.8	226.3	10.569	88.3

Table 11. CO-Net vs Poly-PC under the setting of jointly optimizing ModelNet40, S3DIS, and SUN RGBD.

5. Conclusion

In this work, we introduce CO-Net, a cohesive network that jointly learns multiple point cloud tasks under heterogeneous dataset domains. Leveraging the nested layer-wise processing policy and sign-based gradient surgery, CO-Net attains great performance for all tasks and maintains storage efficiency for model deployment. Moreover, CO-Net is demonstrated to achieve superior performance when generalizing to a new task with minimal task-specific parameter expansion. Extensive experiments illustrate that CO-Net outperforms previous methods by a large margin while retaining fewer total FLOPs and parameters.

Acknowledgement. This work was supported in part by the National Key Research and Development Program of China (2022YFB4702402), in part by Science and Technology Innovation Venture Capital Project of Tiandi Technology Co., LTD. (2022-2-TD-QN009), in part by the National High Technology Research and Development Program of China under Grant 2018YFE0204300, and in part by the National Natural Science Foundation of China under Grant 62273198.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 6

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics*, 37(4), 2018. 7

[3] R Caruana. Multitask learning: A knowledge-based source of inductive bias1. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Citeseer, 1993. 1

[4] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12270–12280, 2021. 1, 6

[5] 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. 2

[6] 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. 2

[7] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8963–8972, 2021. 7

[8] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448, 2021. 6

[9] Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018. 2, 5

[10] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021. 6

[11] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *European conference on computer vision*, pages 544–560. Springer, 2020. 6

[12] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2021. 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[14] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2022. 1, 2, 5

[15] 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. 1

[16] Jae-Han Lee, Chul Lee, and Chang-Su Kim. Learning multiple pixelwise tasks based on loss scale balancing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5107–5116, 2021. 1

[17] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 6, 7

[18] 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

[19] 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. 1, 8

[20] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 2

[21] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 6, 7

[22] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 2, 3, 6, 7

[23] 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. 2

[24] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 7

[25] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4404–4413, 2020. 7

[26] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 6, 7

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 2, 3, 6, 7, 8, 9

[29] Guocheng Qian, Hasan Hammoud, Guohao Li, Ali Thabet, and Bernard Ghanem. Assanet: An anisotropic separable set abstraction for efficient point cloud representation learning. *Advances in Neural Information Processing Systems*, 34:28119–28130, 2021. 7

[30] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. 6, 7, 8

[31] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24:1943–1955, 2021. 2

[32] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24:1943–1955, 2022. 6

[33] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18942–18952, 2022. 6, 7, 8

[34] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019. 2, 5

[35] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 1

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

[37] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point

cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1

[38] Ayan Sinha, Zhao Chen, Vijay Badrinarayanan, and Andrew Rabinovich. Gradient adversarial training of neural networks. *arXiv preprint arXiv:1806.08028*, 2018. 2

[39] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 6

[40] Ziying Song, Haiyue Wei, Caiyan Jia, Yongchao Xia, Xiaokun Li, and Chao Zhang. Vp-net: Voxels as points for 3d object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2023. 2

[41] Ziying Song, Yu Zhang, Yi Liu, Kuihe Yang, and Meiling Sun. Msfyolo: Feature fusion-based detection for small objects. *IEEE Latin America Transactions*, 20(5):823–830, 2022. 2

[42] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 1, 2, 7

[43] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 6

[44] Haiyang Wang, Shaoshuai Shi, Ze Yang, Rongyao Fang, Qi Qian, Hongsheng Li, Bernt Schiele, and Liwei Wang. Rbgnet: Ray-based grouping for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1110–1119, 2022. 7

[45] Li Wang, Ziying Song, Xinyu Zhang, Chenfei Wang, Guoxin Zhang, Lei Zhu, Jun Li, and Huaping Liu. Sat-gcn: Self-attention graph convolutional network-based 3d object detection for autonomous driving. *Knowledge-Based Systems*, 259:110080, 2023. 2

[46] Li Wang, Xinyu Zhang, Ziying Song, Jiangfeng Bi, Guoxin Zhang, Haiyue Wei, Liyao Tang, Lei Yang, Jun Li, Caiyan Jia, et al. Multi-modal 3d object detection in autonomous driving: A survey and taxonomy. *IEEE Transactions on Intelligent Vehicles*, 2023. 2

[47] Shiguang Wang, Tao Xie, Jian Cheng, Xingcheng Zhang, and Haijun Liu. Mdl-nas: A joint multi-domain learning framework for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20094–20104, 2023. 1

[48] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 6

[49] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 9

[50] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Dening Lu, Mingqiang Wei, and Jun Wang. Venet: Voting enhancement network for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3712–3721, 2021. 7

[51] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10447–10456, 2020. 7

[52] Tao Xie, Li Wang, Ke Wang, Ruifeng Li, Xinyu Zhang, Haoming Zhang, Linqi Yang, Huaping Liu, and Jun Li. Farp-net: Local-global feature aggregation and relation-aware proposals for 3d object detection. *IEEE Transactions on Multimedia*, 2023. 2, 7

[53] Tao Xie, Shiguang Wang, Ke Wang, Linqi Yang, Zhiqiang Jiang, Xingcheng Zhang, Kun Dai, Ruifeng Li, and Jian Cheng. Poly-pc: A polyhedral network for multiple point cloud tasks at once. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1233–1243, 2023. 2, 6, 7, 9

[54] Jiahui Yu and Thomas Huang. Autoslim: Towards one-shot architecture search for channel numbers. *arXiv preprint arXiv:1903.11728*, 2019. 1

[55] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1803–1811, 2019. 1

[56] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, pages 702–717. Springer, 2020. 1

[57] 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, 5, 8

[58] Min Zhang, Haoxuan You, Pranav Kadam, Shan Liu, and C-C Jay Kuo. Pointhop: An explainable machine learning method for point cloud classification. *IEEE Transactions on Multimedia*, 22(7):1744–1755, 2020. 2

[59] 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

[60] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, pages 311–329. Springer, 2020. 7

[61] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 1, 7