AutoGPart: Intermediate Supervision Search for Generalizable 3D Part Segmentation

Xueyi Liu[1] Xiaomeng Xu[1] Anyi Rao[2] Chuang Gan[3] Li Yi[1,4]
[1] Tsinghua University [2] The Chinese University of Hong Kong [3] MIT-IBM Watson AI Lab
[4] Shanghai Qi Zhi Institute

Abstract

Training a generalizable 3D part segmentation network is quite challenging but of great importance in real-world applications. To tackle this problem, some works design task-specific solutions by translating human understanding of the task to machine's learning process, which faces the risk of missing the optimal strategy since machines do not necessarily understand in the exact human way. Others try to use conventional task-agnostic approaches designed for domain generalization problems with no task prior knowledge considered. To solve the above issues, we propose AutoGPart, a generic method enabling training generalizable 3D part segmentation networks with the task prior considered. AutoGPart builds a supervision space with geometric prior knowledge encoded, and lets the machine to search for the optimal supervisions from the space for a specific segmentation task automatically. Extensive experiments on three generalizable 3D part segmentation tasks are conducted to demonstrate the effectiveness and versatility of AutoGPart. We demonstrate that the performance of segmentation networks using simple backbones can be significantly improved when trained with supervisions searched by our method.

1. Introduction

Humans parse objects into parts for a deeper understanding of semantics, functionality, mobility as well as the fabrication process. It is natural to consider equipping machines with such part-level understanding. Over the past few years, there has been an increasing interest in 3D part segmentation tasks to support various applications [14, 23, 32, 46] in vision, graphics, and robotics. Thanks to the availability of large-scale 3D part datasets [32, 48] and the development of 3D deep learning techniques, these methods achieve impressive part segmentation results when a test shape comes from the same distribution as the training set. However, they usually suffer from a huge performance drop when it comes to parsing 3D shapes from a novel distribution, *e.g.* from a different semantic category.

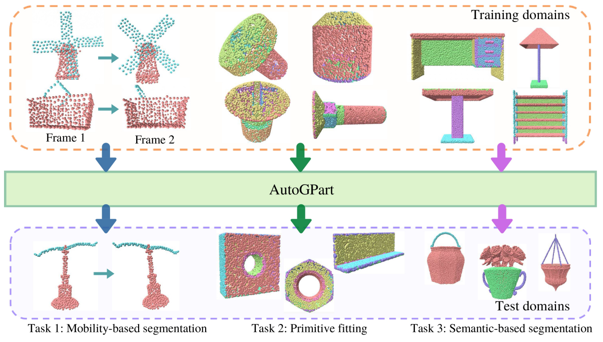
Training domains
Frame 1
Frame 2
AutoGPart
Test domains
Task 1: Mobility-based segmentation
Task 2: Primitive fitting
Task 3: Semantic-based segmentation

Figure 1. *AutoGPart* automatically finds real part cues for different segmentation tasks and supports simple point cloud processing backbones (*e.g.* PointNet++ [35]) to successfully parse shapes from novel categories.

In this work, we focus on generalizable 3D part segmentation tasks. The goal is to learn the essence of parts on some training sets and to be able to generalize well to shapes from novel distributions. A main challenge comes from the versatile cues defining parts. Shape parts can be defined via many cues like geometric primitive fitting [14, 23, 46], rigid motions [12, 13, 47], and semantic prior [27, 29, 32]. The same shape can be segmented into different parts based upon different cues for the seek of various applications. Therefore for a specific application, other than understanding input shapes, a network also needs to figure out the exact cues defining parts from the training data. This is where it becomes tricky. Some of these cues are more salient than others, making them easier to be captured. When these cues heavily correlate with the real part cue in the training domain, they could easily become shortcut features [7, 8, 15] biasing the network and hindering the generalizability, as observed in [29].

To cope with the above challenges, existing works usually focus on a specific type of parts and incorporate part-type priors towards generalizable 3D part segmentation. For example, a modularized design has been adopted to explicitly extract motion flow and the rigid motion of local patches so that generalizable motion-based part segmentation can be achieved in [47]. Similarly, the networks are guided explicitly to extract primitive parameters in order to segment primitive parts in [23, 46]. Finding such explicit part type priors requires a deep understanding of the target task, which might not always be available. Even after a huge amount of trail-and-errors in expert designs, such approaches are still not guaranteed to fully grasp the essence of parts. This is because human understandings might not align well with the way that machines prefer to follow [6, 18, 52].

Another line of works from the machine learning community studies generic domain generalization algorithms to help a network handle out-of-distribution samples. However, without considering any geometric prior which is important for part segmentation tasks, they fail to improve crucial parts of the model but merely focus on generic regularization strategies such as making gradients from multiple domains consistent with each other [30].

In contrast, we present *AutoGPart*, a generalizable 3D part segmentation technique that could be applied to any type of 3D parts. Our key observation is that the generalizability of a 3D part segmentation network is largely hindered by shortcut features [7]. These shortcut features tend to be quite salient and closely correlated with segmentation labels in some training domains. Through finding proper intermediate supervisions that are more closely related with real part cues and loosely correlated with shortcut features, we can downweight the influence of shortcut features by jointly training the network with segmentation and the intermediate supervisions.

We design a parametric model to depict the distribution of possible intermediate supervisions. And we inject geometric priors in the model space such that these supervisions are part-aware and geometrically-discriminative. In addition, we present an automatic way for optimizing the distribution of these supervisions for a supervised part segmentation network. A beam search-like strategy is then applied to greedily generate some suitable supervisions from the distribution. With the additional supervision, task-specific part cues can be automatically discovered without experts' trail-and-errors. Comprehensive experiments on three generalizable 3D part segmentation tasks demonstrate the effectiveness of AutoGPart, including 4.4% absolute Segmentation Mean IoU improvement on mobility-based part segmentation , 4.2% on primitive fitting and 0.7% absolute Mean Recall [29] improvement on semantic-based part segmentation.

To summarize, our contributions are threefold: 1) We present a generic method to improve the generalizability of 3D part segmentation networks via automatically discovered intermediate supervisions and the method is suitable for different part definitions. 2) We design a parametric model to depict the distribution of useful intermediate supervisions with geometric prior encoded. 3) We propose an automatic search algorithm to find the proper intermediate supervision given a specific part segmentation task.

2. Related Work

Domain Generalization. As an important technique to handle out-of-distribution scenarios, existing domain generalization approaches can be categorized into four streams: 1) learning domain invariant features from multiple source domains aiming to minimize the difference between source domains [22, 25, 33], meta-learning [20, 21], and other model-agnostic strategies [30]; 2) data augmentation algorithms to simulate domain shift [55]; 3) ensemble learning techniques that train domain-specific models and uses their ensemble for inference [45, 55]; 4) automated machine learning (AutoML) techniques that aim to automatically search for data augmentation [4, 24] or neural architecture [1, 2] which can achieve optimal generalization performance. Different from previous AutoML strategies, we propose to search for geometric and part-aware features for intermediate supervisions. Such features are invariant across shapes from different distributions, indicating the superiority of our method to improve the generalizability of the network.

3D Part Segmentation. Mobility-based part segmentation, primitive fitting, and semantic-based part segmentation are three important and representative 3D part segmentation tasks, on which we conduct experiments to prove the effectiveness and versatility of AutoGPart . 1) *Mobility-based part segmentation* aims to parse articulated input objects into rigidly moving parts. A number of previous works have devoted into it based on traditional techniques such as clustering and co-segmentation [41, 50], or deep 3D neural networks [13, 47]. 2) *Primitive fitting* addresses the task of clustering input points and fitting them with geometric primitives. Standard solutions include RANSAC [37], region growing [31], supervised learning [14, 23, 46] and unsupervised learning [5, 40]. Recently, [23] proposes an end-to-end neural network that takes point clouds as input and predicts a varying number of primitives. [46] further uses hybrid feature representations to separate points of different primitives. 3) *Semantic-based part segmentation* detects and delineates each distinct object of interest. Conventional approaches rely on manual design on geometric constraints, including K-means [38], graph cuts [9] and spectral clustering [26]. As for learning-based methods [34, 49], although they achieve state-of-the-art performance on seen categories, it is hard for them to parse shapes from unseen categories due to category-variant information that is easy to take than real cues defining parts [29]. Different from previous task-specific methods that employ sophisticated frameworks and delicately designed supervisions, we demonstrate that simple learning frameworks equipped with the intermediate supervisions searched from AutoGPart can achieve comparable or even better performance than existing task-specific designs on the three tasks.

Auxiliary Supervision. Adding auxiliary supervisions is a common strategy to reguralize the learning process for performance improvement. Among them, weight decay is a widely used technique [17, 28]. Besides, multi-task learning [3, 11, 36, 51] trains a network to solve multiple tasks by sharing parameters between different tasks, where the interested branches acquire benefits from other branches. Deeply Supervise [19] and Inception [39] explore how to add auxiliary supervisions on hidden layers to improve the quality of learned low-level features [53, 54]; LabelEnc [10] proposes a new label encoding function that maps ground-truth labels to the latent embedding space to add intermediate supervisions. In this work, we automatically select proper intermediate supervisions for generalizable 3D part segmentation networks. It can be viewed as an approach that utilizes auxiliary supervisions to improve the network's generalizability.

3. Method

We present AutoGPart, a technique to improve the cross-domain generalizability of a supervised 3D part segmentation network by automatically finding useful intermediate supervisions for the network. Then at training time, supervisions found by AutoGPart are introduced in addition to the segmentation supervision to alleviate the influence of shortcut features [7] and capture real part cues that are invariant across different shapes distributions. The resulting network can better generalize to shapes from novel distributions without any additional cost at inference time. Figure 2 shows an example where AutoGPart is plugged into an end-to-end network for primitive segmentation.

To automatically find such supervisions, we design a parametric supervision feature space with geometric prior knowledge encoded (*a parametric intermediate supervision model*, Sec. 3.1). To adapt the model for a specific segmentation network, we optimize its parameters via a "propose, evaluate, update" strategy (Sec. 3.2). After that, a greedy search strategy is utilized to select the optimal supervisions from the optimized model (Sec. 3.3).

3.1. Parametric Modeling for the Supervision Space

Based on the observation that many previous generalizable 3D part segmentation works guide part feature learning via intermediate supervisions calculated from per-point geometric features and ground-truth part labels, we assume that being aware of some of such features can help the network learn real part cues. We propose to automatically search for proper part-aware supervision features for each type of part and add intermediate supervisions to encourage the segmentation network to leverage those features. We then move on to introduce the structural model for supervision features and further the supervision space.

Structural model for supervision features. We define the possible supervision features as the outcome of a tree-structured operation flow (an *operation tree*) taking geometric features and ground-truth part labels as input. It is inspired by the calculation process of hand-crafted part-aware geometric features (*e.g.* rotation matrix, cylinder axis [23], etc.).

Good supervision features could bridge the gap between input point cloud geometry and the output segmentation labels while avoiding shortcuts. And it is crucial to consider how to use ground-truth labels in order to generate such **part-aware and geometrically-discriminative** supervision features. To compute a possible intermediate supervision, instead of treating ground-truth labels as additional input feature vectors, we pre-encode these labels by transferring geometric features of each point to part-aware features. We then use such part-aware features as input to an operation tree directly.

An operation tree transforms input features by operators for the output intermediate supervision feature. Such operators include *grouping operators* (*e.g.* sum, SVD[1], etc.) that summarize a feature set into a point-level feature, point-level *unary operators* (*e.g.* square, double, etc.), and *binary operators* (*e.g.* add, minus, etc.) that encourage feature communications to enlarge and diversify the supervision space.We further introduce some fixed operator-type combinations named *operation cells* such as a unary operator followed by a grouping operator. They are high-frequent operation combinations in the computing process of hand-crafted geometric features [23, 47].

An operation tree is then constructed by connecting operation cells.

Supervision feature space. The supervision feature space consists of all valid operation trees. We set the maximum height of an operation tree to three, thus the supervision feature space is spanned by all possible tree structures and sub-structures of cells in it (Figure 3). To measure the generalization benefit of each supervision feature and to optimize the space toward the optimal intermediate supervisions for a segmentation network, a parametric distribution model ($\mathcal{M}_{\mathcal{T}}(\cdot|\theta)$) is constructed to depict the operation tree space. The subscription $\mathcal{T}$ here indicates

[1] Singular Value Decompose.

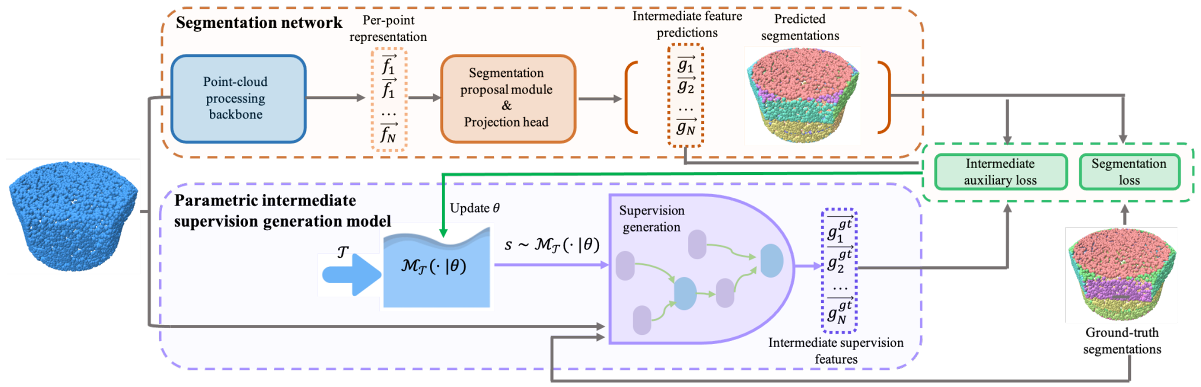
Segmentation network
Per-point representation
Point-cloud processing backbone
Segmentation proposal module & Projection head
Intermediate feature predictions
Predicted segmentations
Intermediate auxiliary loss
Segmentation loss
Parametric intermediate supervision generation model
Update θ
$\mathcal{T}$
$\mathcal{M}_{\mathcal{T}}(\cdot|\theta)$
$s \sim \mathcal{M}_{\mathcal{T}}(\cdot|\theta)$
Supervision generation
Intermediate supervision features
Ground-truth segmentations

Figure 2. An example of applying AutoGPart on an end-to-end segmentation network for the primitive fitting task. AutoGPart builds a learnable parametric supervision model $\mathcal{M}_{\mathcal{T}}(\cdot|\theta)$ that can be optimized in order to find proper intermediate supervisions for a generalizable 3D part segmentation network. A "propose, evaluate and update" strategy is adopted to learn θ. In each circle, an operation tree is sampled from $\mathcal{M}$, which is then used to calculate part-aware geometric features for each point. The generalization ability of the proposed supervision is evaluated and the resulting score is further used to update θ in the updating stage. Light purple lines imply the supervision generation process and green for the parameter updating process.

that it is constructed based on prior knowledge of the task set $\mathcal{T}$, θ denotes parameters introduced in the distribution model. In practice, a marginal distribution is introduced for the grouping operator of the root cell. Then, conditional distributions are introduced for children operators and leaf part-aware features conditioned on their parents.

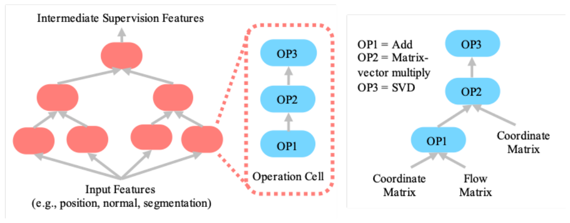
Intermediate Supervision Features
OP3
OP2
OP1
Operation Cell
Input Features (e.g., position, normal, segmentation)
OP1 = Add
OP2 = Matrix-vector multiply
OP3 = SVD
Coordinate Matrix
Flow Matrix

Figure 3. **Left side:** **Left side:** Supervision feature space. **Right Side:** An example for the operation tree whose output feature is the rotation matrix calculated by the Orthogonal Procrustes algorithm. Operation cell abstraction is not drawn on it.

3.2. Learning Task-Conditioned Supervision Distribution

To optimize the distribution parameters θ for a segmentation network, we adopt a "propose, evaluate and update" strategy.

Supervision proposal. Since a supervision feature is modeled by its generation process in our model, an operation tree depicting the generation of a supervision feature is sampled from the parametric model $\mathcal{M}_{\mathcal{T}}(\cdot|\theta)$ to generate a supervision feature. After that, the input part-aware geometric feature matrices are passed through the sampled operation tree to generate the corresponding supervision feature s. Then the ground-truth intermediate supervision feature $\vec{g_i}^{gt}$ for each point i is calculated by passing its input features through s.

Supervision evaluation. The cross-domain generalization ability of the proposed supervision feature is estimated under a simulated domain-shift setting. Firstly, the training domain is split into different sub-domains. After that, the OOD performance of the sampled supervision feature is estimated by the average generalization error calculated on all of those train-validation splits created by learning one sub-domain out for validation. The network is trained together with the proposed supervision s and the segmentation task-related supervision to evaluate the generalization benefit of s.

Supervision space optimization. The parameter θ is updated by the probability density value of the generated supervision ($\mathcal{M}_{\mathcal{T}}(s|\theta)$) and its estimated generalization score. The updating strategy is derived from the REINFORCE [44] algorithm. Simply updating parameters of conditional distributions involved in the generation process of s via REINFORCE equals to updating the joint distribution by REINFORCE.

3.3. Greedy Supervision Selection

Though we can select a single supervision feature from the supervision model after the optimization (e.g. the feature with the highest probability), we wish to select multiple features to use for better generalizability enhancement.

To be more specific, we adopt a greedy search strategy to select several supervisions (one to three, in our practice) to use further for the segmentation network. The greedy search aims to choose an optimal supervision set step-by-step. It starts from a set containing single supervisions sampled from the optimized supervision space. The generalization ability of such supervision features are then estimated. After that, supervisions with top performance are kept and further used to construct a set containing two supervision pairs. Similarly, the performance of supervision combinations are estimated with a new set constructed by coupling top supervisions in each following step. Finally, the supervision combination achieved the best performance is then selected to use further.

4. Experiments

We evaluate the effectiveness of the proposed AutoGPart in searching for suitable intermediate supervisions and improving the domain generalization ability of a segmentation network on three 3D part segmentation tasks: mobility-based part segmentation [13, 47], primitive fitting [14, 23, 46] and semantic-based part segmentation [29].

For each task, we set a default segmentation network to evaluate AutoGPart on, which is a point-cloud processing backbone for representation learning followed by a classification-based segmentation module to propose segmentations. Two backbones are used in experiments, namely PointNet++ [35] and DGCNN [43]. We denote such two segmentation networks as "PointNet++" and "DGCNN" directly for simplicity. For each task, we demonstrate that a simple 3D point-cloud segmentation network trained with intermediate supervisions searched by AutoGPart and segmentation task-related supervisions is able to perform better than task-specific models proposed in previous works as well as networks trained by generic domain generalization strategies.

4.1. Mobility-based Part Segmentation

Datasets. Three datasets are used in the task: training dataset, auxiliary training dataset that is only used in the supervision search stage to help simulate domain-shift, and the out-of-domain (OOD) test dataset. The training dataset is created from [48], containing 15,776 shapes from 16 categories. The auxiliary training dataset is created from PartNet [32], containing 5,662 shapes from 4 categories different from those used in the training dataset. The test dataset used is the same as the one used in [47], which is created from [13], containing 875 shapes covering 175 objects from 23 categories.

Experimental settings. We evaluate the effectiveness of AutoGPart on both PointNet++ and DGCNN. Metric used for this task is Segmentation Mean IoU (MIoU). To apply AutoGPart on this task, we add a flow estimation module ahead of the segmentation network. We compare our method with both task-specific baselines [41, 47, 50], and task-agnostic methods [20, 55]. Deep Part Induction [47] is trained on the same training dataset using the same settings as those used for our models. All learning-based models use only one forward flow estimation and segmentation proposal pass with no iteration between them.

Table 1. Experimental results on the mobility-based segmentation task. For abbreviations used, In/Out-of-dist. refers to In/Out-of-distribution performance; "PN++" denotes "PointNet++"; "JLC" means "JLinkage clustering"; "SC" means "Spectral Clustering". Subscriptions indicate the used backbones.

Method	In-dist.	Out-of-dist.
PointNet++	86.4	61.0
DGCNN [43]	88.3	68.1
MixStyle$_{\text{PN++}}$ [55]	86.6	63.4
MixStyle$_{\text{DGCNN}}$ [55]	83.1	69.5
Meta-learning$_{\text{PN++}}$ [20]	71.3	63.4
Meta-learning$_{\text{DGCNN}}$ [20]	76.5	69.4
JLC [50]	N/A	67.3
SC [41]	N/A	69.4
Deep Part Induction [47]	84.8	64.4
AutoGPart$_{\text{PN++}}$	87.2	66.5
AutoGPart$_{\text{DGCNN}}$	83.1	**73.8**

Experimental results. The performance of AutoGPart and baseline models are summarized in Table 1. Based on the table, we can make the following observations: 1) AutoGPart can find useful intermediate supervisions that can significantly improve the generalization ability of both PointNet++ and DGCNN (*e.g.* 4.9% absolute MIoU improvement on OOD-test set for PointNet++). This can verify the effectiveness of adding intermediate supervisions to boost the generalization ability of segmentation networks and the ability of AutoGPart to find such supervisions. 2) AutoGPart can outperform all task-specific models by a large margin, including traditional methods (*e.g.* JLinkage clustering) and learning based strategies such as Deep Part Induction. This may echo the proposed assumption that hand-crafted supervisions may not be optimal ones for the task due to the misalignment between human knowledge and machine understanding. 3) AutoGPart can outperform all task-agnostic strategies when using the same point-cloud processing backbone. A possible reason is that geometric prior knowledge which is quite useful to solve the task inherently is carefully considered in the design of AutoGPart , while task-agnostic strategies are not aware of such prior knowledge.

4.2. Primitive Fitting

Dataset. Dataset used in this task is the same as the one used in [23]. However, instead of using the provided data splitting method directly, we re-split the dataset into 4 subsets according to shapes such that it is more suitable to test the cross-domain generalization ability of a model. Three subsets out of them, containing 13,528 shapes in total, are used for training, including the supervision search stage and regular training stage for supervision evaluation.

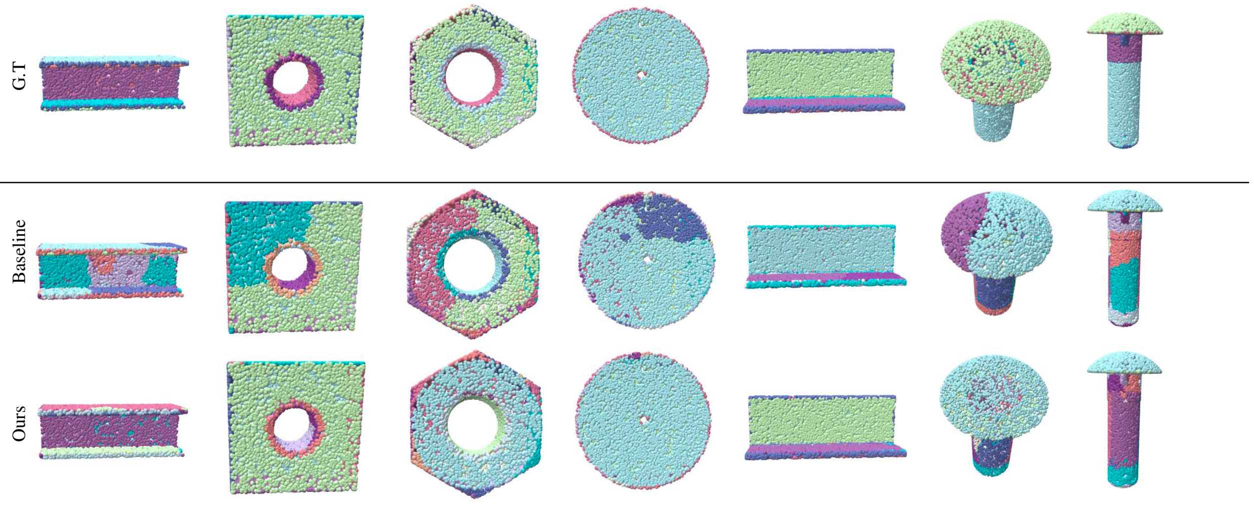

Figure 4. Segmentation results visualization for the primitive fitting task. "Ours" denotes the model "AutoGPart$_{\text{HPNet}}$" and "Baseline" denotes the model "HPNet" in Table 2.

The left one subset, containing 3,669 shapes, is used for the out-of-domain test. For all compared baselines, we test their performance on the re-split dataset for a fair comparison.

Experimental settings. We evaluate the effectiveness of AutoGPart on both PointNet++ and DGCNN. Metric used for this task is MIoU. We compare our method with both task-specific baselines [23, 46] and task-agnostic methods [20, 30, 55].

Among them, HPNet [46] adopts a clustering-based two-stage segmentation network for feature learning and segmentation proposal. Based on the observation that the high performance achieved by HPNet is largely powered by its clustering-based segmentation module and the synergies between the added supervisions and the clustering module, we conduct the following two experiments to fairly compare AutoGPart with HPNet, since the classification-based segmentation module used in default settings of AutoGPart is not that representative compared with Mean-Shift clustering used in HPNet: 1) Plug AutoGPart in HPNet's first learning stage to evaluate the effectiveness of AutoGPart in finding useful features that can regularize this learning stage. 2) Replace the clustering-based segmentation proposal module used in HPNet with a classification-based module to compare the effectiveness of supervisions introduced in HPNet and those searched by AutoGPart under a same segmentation network. Resulted models are denoted as "AutoGPart$_{\text{HPNet}}$" and "HPNet*" respectively in Table 2.

Experimental results. The performance of AutoGPart and other baseline models are summarized in Table 2. We can make the following observations from Table 2: 1) AutoGPart can find useful intermediate supervisions to improve the generalization ability of a simple segmentation network for the primitive fitting task using either PointNet++ or DGCNN as the backbone, similar with that observed for mobility-based part segmentation task. 2) For models using classification-based segmentation modules, AutoGPart can boost a simple segmentation network's performance better than all task-specific models such as SPFN. A possible reason is that the added supervisions, though believed useful by human to help the network learn a correct solution inherently, may not align well with the way preferred by machines to solve the task. Thus, those hand-crafted supervisions may not be effective enough to train a generalizable network. Besides, networks may even use shortcut features to optimize those supervisions. For models using clustering-based segmentation modules, AutoGPart$_{\text{HPNet}}$ can achieve better generalization performance than HPNet by adding some intermediate supervisions in its first learning stage. It is probably because that the added supervisions searched by AutoGPart can help the network use more real cues to learn per-point features in this stage. 3) AutoGPart can help better improve the generalization ability of 3D part segmentation networks compared with task-agnostic methods.

The absolute improvement that AutoGPart adds on HPNet is not as significant as PointNet++ or DGCNN, for which we want to emphasize two points as follows: 1) The high performance achieved by HPNet is largely benefit from the clustering-based segmentation module (*i.e.* 69.6% MIoU on OOD-test set achieved by HPNet* v.s. 79.5% achieved by HPNet). However, such a segmentation proposal process is quite **time-consuming**, where about 40 hours are needed to segment all test shapes, while a classification-based one only needs no more than 40 seconds. 2) For classification-based segmentation networks, intermediate supervisions searched by AutoGPart are clearly better than that used in HPNet

Table 2. Experimental results on the primitive fitting task. For abbreviations used, In/Out-of-dist. refers to In/Out-of-distribution performance; "PN++" denotes "PointNet++". Subscriptions indicate the used backbones; "GS" means "Gradient Surgery". "HPNet" and "AutoGPart$_{\text{HPNet}}$" use clustering-based segmentation modules. Others use classification-based modules.

Method	In-dist.	Out-of-dist.
PointNet++	81.5	71.6
DGCNN	93.6	68.0
GS$_{\text{PN++}}$ [30]	90.7	70.3
GS$_{\text{DGCNN}}$ [30]	92.6	71.6
Meta Learning$_{\text{PN++}}$ [20]	65.0	68.5
Meta-learning$_{\text{DGCNN}}$ [20]	67.1	69.3
MixStyle$_{\text{PN++}}$ [55]	92.1	70.9
MixStyle$_{\text{DGCNN}}$ [55]	93.7	71.4
SPFN [23]	94.4	72.3
HPNet* [46]	93.9	69.6
HPNet [46]	N/A	79.5
AutoGPart$_{\text{PN++}}$	86.3	76.5
AutoGPart$_{\text{DGCNN}}$	94.2	73.4
AutoGPart$_{\text{HPNet}}$	N/A	**80.4**

by comparing the performance of AutoGPart$_{\text{DGCNN}}$ with HPNet* (*i.e.* 73.4% v.s. 69.6% MIoU on OOD-test set). The significant effectiveness of AutoGPart in improving the generalization ability of an end-to-end trained model is of larger practical value.

4.3. Semantic-based Part Segmentation

Dataset. We use the same dataset provided by [29] as well as the train-test data splitting strategy stated in the paper.

Experimental Settings. The evaluation metric used in this task is the Mean Recall value, the same as the one used in [29]. Values reported in Table 3 are the average Mean Recall scores over all test categories. We compare AutoGPart with four task-specific methods [16, 29, 42, 49], and two task-agnostic approaches [20, 55].

Experimental results. The performance of AutoGPart and other baseline models are summarized in Table 3. Different from the above two tasks, where the criterion about segmenting a shape is obvious such as rigid motions, it is not that clear what part cues are useful for this task, meaning not enough prior knowledge to guide a learning-based network's design. Thus, the role of ground-truth geometric features is not that important in the previous task-specific designs [29, 42, 49]. However, their resulting models tend to perform not that well when parsing shapes from novel categories, as shown in Table 3, probably due to the influence of shortcut features. However, our method can help improve the generalization ability of a simple segmentation network to a level comparable to the two-stage learning-based method [29] as well as traditional segmentation methods

Table 3. Experimental results on the semantic-based part segmentation task. For abbreviations used, In/Out-of-dist. refers to In/Out-of-distribution performance. The backbone used in AutoGPart is PointNet++.

Method	In-dist.	Out-of-dist.
MixStyle [55]	34.4	30.7
Meta-learning [20]	35.1	29.7
PartNet-InstSeg [32]	33.9	26.7
SGPN [42]	25.0	20.2
GSPN [49]	23.5	28.7
Learning to Group [29]	35.2	32.0
WCSeg [16]	29.8	33.2
AutoGPart	35.7	**33.9**

such as WCSeg [16]. It demonstrates the usefulness of ground-truth part-aware features in providing real cues for a generalizable segmentation network; And also indicates the superiority of AutoGPart to find such useful features for a task where human prior knowledge is not that available or hard to be translated to guide a network's learning process.

4.4. Qualitative Evaluation

We visualize the segmentation results of "AutoGPart$_{\text{HPNet}}$" and "HPNet" in Table 2 on the Primitive Fitting task for a intuitive comparison and understanding w.r.t. the network's generalization ability on OOD shapes. Figure 4 shows that AutoGPart$_{\text{HPNet}}$ can achieve better segmentation performance on shapes from a novel distribution, i.e., having a relatively large primitive-type distribution gap from that of training shapes. A possible reason is that the added intermediate supervisions in the first stage of HPNet can help the model learn per-point features relying more on real part cues, thus less sensitive to the primitive type of each part as well as the primitive-type distribution of the shape.

Besides, we draw an intermediate supervision feature searched from our designed supervision space in Figure 6 for an intuitive understanding towards the property of our searched feature. It can be seen that the searched feature is not only discriminative across different parts, but also changes continuously within one part. Such property may make it be more friendly for the network learning real part cues avoiding shortcut features that only exist in shapes from training distributions.

5. Ablation Study

In our method, we construct an intermediate supervision space and search for useful supervisions from it to increase the generalizability of a segmentation network. In this section, we ablate our method by replacing some crucial designs with other possible alternatives to analyze the effect of those parts.

Table 4. Ablation study w.r.t. reward function design and supervision space design. For abbreviations used, "Arch." refers to "Architecture"; "PN++" denotes "PointNet++"; In/Out-of-dist. refers to In/Out-of-distribution performance.

Ablation	Arch.	In-dist.	Out-of-dist.
/	PN++	87.2	**66.5**
−Cross Validation		85.6	64.6
−Gap		90.6	64.9
Less operants		87.5	63.1
Less unary operators		86.8	64.1
Less binary operators		83.0	65.1
Less tree height		82.1	63.5
/	DGCNN	83.1	**73.8**
−Cross Validation		84.5	73.1
−Gap		89.4	70.4
Less operants		92.1	70.6
Less unary operators		89.3	69.5
Less binary operators		89.7	71.6
Less tree height		89.4	71.8

Reward function design. In AutoGPart , we adopt the average generalization gap over all train-validation splits as the reward value to estimate the effectiveness of the selected supervision on improving the model's generalization ability. The intuition is that the cross-validating the generalization gap of the searched supervision feature could probably give its generalizability a better estimation. In Table 4, we validate the superiority of the designed reward function by replacing it with 1) generalization gap on a single split (denoted as "−Cross validation"), and 2) average performance on validation sets over all splits (denoted as "−Gap").

Supervision space design. In our method, we use a large supervision feature space to increase the diversity of features in it and also to enable the model a high freedom to select its preferred features by itself at the same time. To demonstrate its benefit, we downsize the supervision space by downsizing the input part-aware geometric feature set, the unary operator set and the binary operator set respectively and then evaluate the effectiveness of the searched supervisions. Experimental results prove the superiority of using a large supervision space (see Table 4).

Supervision selection strategy. In our method, we use a greedy search-like strategy to select supervision features for future use based on the intuition that such a strategy can help us find a good supervision feature combination from the optimized space. To demonstrate the effectiveness of the greedy supervision selection strategy used in our method, we try to ablate it and consider the following two alternatives: 1) Select top $K (1 \leq K \leq 3)$ features from the supervision feature set ordered by the probability density value. Results are summarized in Table 5. 2) Choose features from the optimized distributions randomly. Ten random features are sampled and their results are summarized in Figure 5.

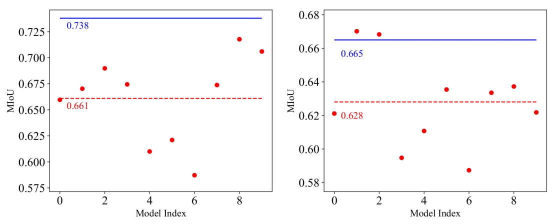

Figure 5. Ablation study w.r.t. supervision selection strategy. Ten supervision features are randomly selected from the optimized distributions and evaluated. Left for PointNet++ and right for DGCNN.

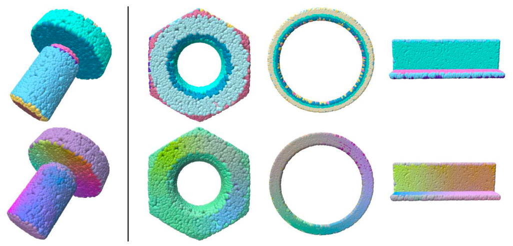

Figure 6. Visualization for one searched intermediate supervision feature from our designed feature space. Upper: Segmentations; Lower: Searched feature (converted to RGB values). The left most shape is the one from the training dataset, while the other three are from the test dataset.

Table 5. Ablation study w.r.t. the supervision selection strategy. For abbreviations used, "Arch." refers to "Architecture"; "PN++" denotes "PointNet++"; In/Out-of-dist. refers to In/Out-of-distribution performance.

Ablation	Arch.	In-dist.	Out-of-dist.
/	PN++	87.2	**66.5**
Top1		87.1	65.6
Top2		81.5	64.5
Top3		85.8	62.8
/	DGCNN	83.1	**73.8**
Top1		92.1	70.1
Top2		89.8	67.8
Top3		90.7	71.6

6. Discussion and Conclusion

In this paper, we propose to automatically search for proper intermediate supervisions for generalizable 3D part segmentation networks.

Although experimental results can prove the effectiveness and versatility of AutoGPart to some extend, there are still many directions worth exploring by the community in the future: 1) Search intermediate supervisions and the backbone's structure at the same time. Synergies between backbone architecture and intermediate supervisions may be found since different backbones may prefer different features. 2) How to add intermediate supervisions. Directly predicting a ground-truth feature may not be suitable for each feature. Moreover, synergies between supervisions features and how to supervise the network to learn such features may be discovered.

References

[1] Haoyue Bai, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8320–8329, 2021. 2

[2] Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. AdaNet: Adaptive structural learning of artificial neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 874–883. PMLR, 06–11 Aug 2017. 2

[3] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 3

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 2

[5] Hao Fang, Florent Lafarge, and Mathieu Desbrun. Planar shape detection at structural scales. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2973, 2018. 2

[6] Christina M Funke, Judy Borowski, Karolina Stosio, Wieland Brendel, Thomas SA Wallis, and Matthias Bethge. The notorious difficulty of comparing human and machine perception. *arXiv e-prints*, pages arXiv–2004, 2020. 2

[7] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1, 2, 3

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1

[9] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3d mesh analysis. In *ACM SIGGRAPH Asia 2008 papers*, pages 1–12. 2008. 2

[10] Miao Hao, Yitao Liu, Xiangyu Zhang, and Jian Sun. Labelenc: A new intermediate supervision method for object detection. In *European Conference on Computer Vision*, pages 529–545. Springer, 2020. 3

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[12] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3d shapes via subspace clustering. In *Computer graphics forum*, volume 31, pages 1703–1713. Wiley Online Library, 2012. 1

[13] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 1, 2, 5

[14] Jingwei Huang, Yanfeng Zhang, and Mingwei Sun. Primitivenet: Primitive instance segmentation with local primitive embedding under adversarial metric. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15343–15353, 2021. 1, 2, 5

[15] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019. 1

[16] Oliver Van Kaick, Noa Fish, Yanir Kleiman, Shmuel Asafi, and Daniel Cohen-Or. Shape segmentation by approximate convexity analysis. *ACM Transactions on Graphics (TOG)*, 34(1):1–11, 2014. 7

[17] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992. 3

[18] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019. 2

[19] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. PMLR, 2015. 3

[20] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 5, 6, 7

[21] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455, 2019. 2

[22] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. 2

[23] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2652–2660, 2019. 1, 2, 3, 5, 6, 7

[24] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Dada: Differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*, 2020. 2

[25] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 2

[26] Rong Liu and Hao Zhang. Segmentation of 3d meshes through spectral clustering. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pages 298–305. IEEE, 2004. 2

[27] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3496–3504, 2017. 1

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[29] Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to group: A bottom-up framework for 3d part discovery in unseen categories. *arXiv preprint arXiv:2002.06478*, 2020. 1, 2, 3, 5, 7

[30] Lucas Mansilla, Rodrigo Echeveste, Diego H. Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6630–6638, October 2021. 2, 6, 7

[31] David Marshall, Gabor Lukacs, and Ralph Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):304–314, 2001. 2

[32] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 1, 5, 7

[33] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 10–18, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. 2

[34] 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. 2

[35] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 5

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 3

[37] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007. 2

[38] Shymon Shlafman, Ayellet Tal, and Sagi Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Computer graphics forum*, volume 21, pages 219–228. Wiley Online Library, 2002. 2

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[40] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2

[41] Dimitrios Tzionas and Juergen Gall. Reconstructing articulated rigged models from rgb-d videos. In *European Conference on Computer Vision*, pages 620–633. Springer, 2016. 2, 5

[42] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018. 7

[43] 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. 5

[44] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. 4

[45] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014. 2

[46] Siming Yan, Zhenpei Yang, Chongyang Ma, Haibin Huang, Etienne Vouga, and Qixing Huang. Hpnet: Deep primitive segmentation using hybrid representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2753–2762, October 2021. 1, 2, 5, 6, 7

[47] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*, 2018. 1, 2, 3, 5

[48] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 1, 5

[49] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019. 2, 7

[50] Qing Yuan, Guiqing Li, Kai Xu, Xudong Chen, and Hui Huang. Space-time co-segmentation of articulated point cloud sequences. In *Computer Graphics Forum*, volume 35, pages 419–429. Wiley Online Library, 2016. 2, 5

[51] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017. 3

[52] Zijian Zhang, Jaspreet Singh, Ujwal Gadiraju, and Avishek Anand. Dissonance between human and machine understanding. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–23, 2019. 2

[53] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic

segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–284, 2018. 3

[54] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3

[55] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021. 2, 5, 6, 7