

Mitigating Geometric Degradation in Fast DownSampling via FastAdapter for Point Cloud Segmentation

Shuofeng Sun¹, Haibin Yan^{1*}

Beijing University of Posts and Telecommunications¹

Abstract

Farthest Point Sampling (FPS) is widely used in existing point-based models because it effectively preserves structural integrity during downsampling. However, it incurs significant computational overhead, severely impacting the model's inference efficiency. Random sampling or grid sampling is considered **faster downsampling methods**; however, these fast downsampling methods may lead to the loss of geometric information during the downsampling process due to their overly simplistic and fixed rules, which can negatively affect model performance. To address this issue, we propose FastAdapter, which aggregates local contextual information through a small number of anchor points and facilitates interactions across spatial and layer dimensions, ultimately feeding this information back into the downsampled point cloud to mitigate the information degradation caused by fast downsampling methods. In addition to using FastAdapter to enhance model performance in methods that already employ fast downsampling, we aim to explore a more challenging yet valuable application scenario. Specifically, we focus on pre-trained models that utilize FPS, embedding FastAdapter and replacing FPS with random sampling for lightweight fine-tuning. This approach aims to significantly improve inference speed while maintaining relatively unchanged performance. Experimental results on ScanNet, S3DIS, and SemanticKITTI demonstrate that our method effectively mitigates the geometric information degradation issues caused by fast downSampling.

1. Introduction

With the advancement of deep learning models and sensor technology, there is a growing interest among researchers in using neural networks to recognize point cloud data [7, 8, 25, 26, 31, 32]. One important issue is that some practical 3D applications, such as autonomous driving and augmented reality, demand high inference speeds from neural networks. However, due to the unordered and irregular na-

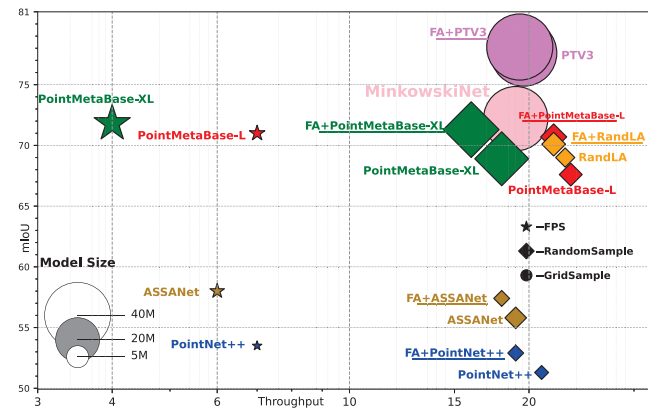


Figure 1. **Model Comparison by Size, mIoU and Throughput(inference)**. We visualized the size, mIoU, and throughput of different models on ScanNet. First, for most point-based models that use FPS, it is evident that FPS generally results in slower inference speeds. However, while replacing FPS with random sampling significantly improves inference speed, it leads to severe performance degradation. By embedding our FastAdapter (FA+XXXX), we can effectively mitigate this performance degradation issue. Secondly, incorporating FastAdapter into models that already utilize the fast downSample (like RandLA with random-sample and PTV3 with gridsample) method results in a significant performance improvement.

ture of point cloud data, point-based models typically use custom sampling and grouping operators instead of traditional convolutional neural networks to extract local features, which often incurs significant computational overhead.

The downsampling method is a part of the computational resource overhead. Farthest point sampling (FPS) is the most commonly used downsampling method, which involves selecting the point in the point cloud that is farthest from the already selected points as the new sampling point. This ensures a relatively regular density distribution and preserves the integrity of local structures during the down-sampling process. However, the need to compute the distance matrix between points incurs significant computational overhead ($O(N^2)$), which results in slow in-

*Corresponding author

ference speed, which affects its application on large-scale point clouds.

In contrast, random sampling is highly efficient, with constant time complexity, making it suitable for large-scale point cloud scenarios. Similarly, GridSample was proposed in PTV2 [40] as a replacement for FPS. It adopts a voxel-based downsampling approach, which also brings about a significant increase in inference speed. For convenience, we collectively refer to these simple and efficient downsampling methods as **fast downSampling**. However, due to their straightforward and fixed rules, these fast downSampling methods are likely to lose geometric information during the downsampling process in complex point cloud scenarios, compromising the integrity of the geometric structure (We will provide detailed explanations in the supplementary materials). This issue of information degradation limits the application of fast downSampling, particularly the simplest and most efficient random sampling approach. An example, as shown in Fig. 1, illustrates that both classic models like PointNet++ and state-of-the-art models such as PointMetaBase-L/XL experience significant performance degradation when FPS is replaced with random sampling. Some methods have attempted to address the issues arising from random sampling. RandLA [11] introduces a multi-layer local aggregation module to expand the receptive field and supplement more local information. APP-Net [20] uses anchor points to aggregate local information and propagates features to the input point cloud. However, it lacks refinement of the features from the anchor points, which can lead to errors in the simply aggregated local features. More importantly, previous methods require the design of unique network architectures to address the issues brought about by random sampling, making them not universally applicable to other models.

Therefore, in this paper, we propose FastAdapter, a universal adapter mechanism designed to mitigate the information degradation issues associated with fast downsampling methods. Specifically, FastAdapter consists of the P2A (Point2Anchor) Aggregator and the A2P (Anchor2Point) Adapter. For P2A Aggregation, we first establish fixed anchor points in the scene. Each point in the point cloud is then assigned to the nearest anchor point, and its features are injected into the anchor point to aggregate local contextual information. To refine the features aggregated at the anchor points, we facilitate information exchange between the anchor points across both spatial dimensions and different layers. Then, in the A2P Adapter, the features of the anchor points are adaptively sent back to the downsampled point cloud based on geometric relationships. This process helps to supplement the lost local features and mitigates the information degradation problem.

We want to emphasize that our method is independent of the model and sampling method, allowing it to be uni-

versally embedded into existing approaches. An intuitive idea for utilizing FastAdapter is to embed it into models designed for fast downSampling (like RandLA with randomsample or PTV3 [41] with GridSample). By doing so, FastAdapter can further alleviate the issues of geometric information degradation, effectively enhancing the overall performance of the model. However, a more practical challenge is that most currently trained point-based models are designed for FPS, such as the widely used PointNet++. Therefore, we propose to fix the weights of these models and embed FastAdapter, simultaneously replacing FPS with random sampling for fine-tuning. This approach aims to preserve the original knowledge learned by the model while effectively mitigating the geometric information degradation issues brought about by random sampling and improve inference speed several times. Some examples are shown in Fig. 1. We primarily focus on the point cloud segmentation task and conducted experiments on ScanNet, S3DIS, and SemanticKITTI. We simultaneously tested various point-based models, including those utilizing Fast DownSampling (RandLA with randomsample and PTV3 with gridsample) and those designed with FPS (PointNet++ with FPS and PointMetaBase-L/XL with FPS). In summary, FastAdapter is a universal method that can effectively address the issue of key information loss associated with fast downSampling. We hope that this work will encourage researchers to focus on random sampling or other fast downsampling methods in order to achieve faster point cloud analysis.

Our contributions can be summarized as follows:

1. We propose FastAdapter, an adapter mechanism designed to mitigate the information degradation problem associated with Fast DownSampling methods, which can be universally embedded into existing point-based models.
2. We propose the P2A Aggregator and A2P Adapter. The former gradually injects the features of the input point cloud into the anchor points to aggregate local information, while the latter propagates the local information from the anchor points back to the input point cloud to supplement the local structural features lost due to fast downsampling methods.
3. FastAdapter is tested on multiple segmentation datasets and models, and the results verify its effectiveness.

2. Related Work

Voxel-based Methods. Due to the irregularity of points, CNN is difficult to be directly applied to point cloud data. Therefore, some works [6, 8, 21, 22, 42, 43, 46] voxelize point clouds into regular grids, and then mature 2D networks can be used to identify point clouds. However, due to the voxelization of point clouds, many empty voxels are present in the scene. Traditional 3DCNN processes these empty voxels, leading to unnecessary computational over-

head. Therefore, to accelerate voxel-based models, sparse convolution [4, 9, 24] is introduced, which is applied only to non-empty voxels, significantly improving the inference speed.

Point-based Methods. Point-based models operate directly on the point cloud data, which prevents missing information caused by operations such as voxelization or projection. PointNet [25] first proposes to use symmetric aggregation function to solve the disorder problem of point cloud, and then PointNet++ [26] replaces CNN to capture local features on irregular point cloud by custom downsampling and grouping operator. Many subsequent works [11, 16, 20, 30, 31, 34, 35, 40, 41, 44] design more complex local feature extraction modules based on PointNet++. However, with the development of models, more and more people pay attention to how to identify point clouds more efficiently. PTV2 [40] designs grouped vector attention, which greatly reduces the computational overhead of the attention mechanism through the grouping strategy. APPNet [20] designs the push-pull operator, which assigns each point to only one neighborhood, reducing the overhead of repeated feature extraction. RandLANet [11] uses random sampling and a specialized architecture.

Transfer Learning. Transfer learning refers to the process of adapting a model to a new data distribution. Full fine-tuning updates all parameters of the model without the need for carefully designed tuning methods. While this approach is simpler, it carries the risk of losing the knowledge that was acquired during the initial training. Prompt Tuning [12, 19, 36, 37, 45] guides the fine-tuning of a model by augmenting the input with additional prompt information. In contrast, Adapter Tuning [3, 10, 13–15, 45, 47] adjusts the model’s output features by inserting lightweight Adapter modules into the model, making them more aligned with the target distribution.

3. Proposed Method

In this section, we first outline FastAdapter’s task in 3.1, which is how to achieve faster point cloud segmentation by using FastAdapter and fast downSampling. Then, we introduce the P2A Aggregator and A2P Adapter in 3.2 and 3.3, respectively. Finally, we present the overall training loss in 3.4. The overall framework of FastAdapter can be seen in Figure 2.

3.1. Preliminary

1) *Background:* Existing point-based models generally follow the steps below when extracting local features from points:

$$p^c, f^c = \phi_d(p, f) \quad (1)$$

$$p_N, f_N = \phi_t(\phi_g(p^c, f^c, p, f)) \quad (2)$$

$$\hat{p}, \hat{f} = \phi_r(p_N, f_N) \quad (3)$$

where $p \in R^{n \times 3}$ and $f \in R^{n \times C}$ represents the coordinates and features of n points. Then, ϕ_d means the downsampling methods, which then selects m center points $\{(p^c \in R^{m \times 3}, f^c \in R^{m \times C})\}$ from p and f . After that, the point-based model uses ϕ_g to construct local neighborhoods for each center point and extracts the features of each neighboring point using ϕ_t . Finally, it aggregates the local features into the center points using ϕ_r .

2) *Problem Statement:* FPS is widely used in most point-based models; however, its computational overhead is significant, impacting the model’s inference efficiency. In contrast, methods like random sampling and gridsample, which have simple and fixed rules, offer higher efficiency. We refer to these sampling methods as fast downSampling.

Above all, existing point-based models can be classified into two categories: one is $\phi_d = FPS$, and the other is $\phi_d \in \{RandomSample, GridSample\}$. Therefore, FastAdapter has two main tasks: (1) For pretrained-models that $\phi_d = FPS$, since they are not designed with architectures for fast downSampling, directly replacing FPS with fast downSampling methods, such as random sampling, can lead to severe performance degradation. Our goal is to embed FastAdapter and fine-tune the pre-trained model so that it can adapt to random sampling, significantly improving inference speed while maintaining performance relatively unchanged. The reason for choosing random sampling is that it presents greater challenges and can better demonstrate the effectiveness of our method. (2) For models that have already utilized fast downSampling ($\phi_d \in \{RandomSample, GridSample\}$), their architectural design can somewhat mitigate the information loss caused by fast downSampling. Therefore, we aim to embed FastAdapter and train from scratch to demonstrate that FastAdapter can complement such models effectively.

It is important to emphasize that for the first task, we do not train from scratch but instead fine-tune the pre-trained models. This is because these models are designed for FPS, and training with FPS yields better results. By simply using FastAdapter and random sampling for lightweight fine-tuning, we can inherit competitive performance. And the second task can be regarded as a special case of the first task, where we fine-tune the weights of all the models. To this end, we will subsequently introduce the method based on the settings of the first task, including the process shown in Fig. 2. Therefore, we define the original model that uses FPS as M_{FPS} , and the model that incorporates FastAdapter and undergoes fine-tuning with random sampling as M_{FA} .

3.2. P2A Aggregator

The purpose of P2A Aggregator is to gradually aggregate local features into anchor points and obtain well-represented local structures through feature refinement so that the A2P Adapter can supplement the structural features

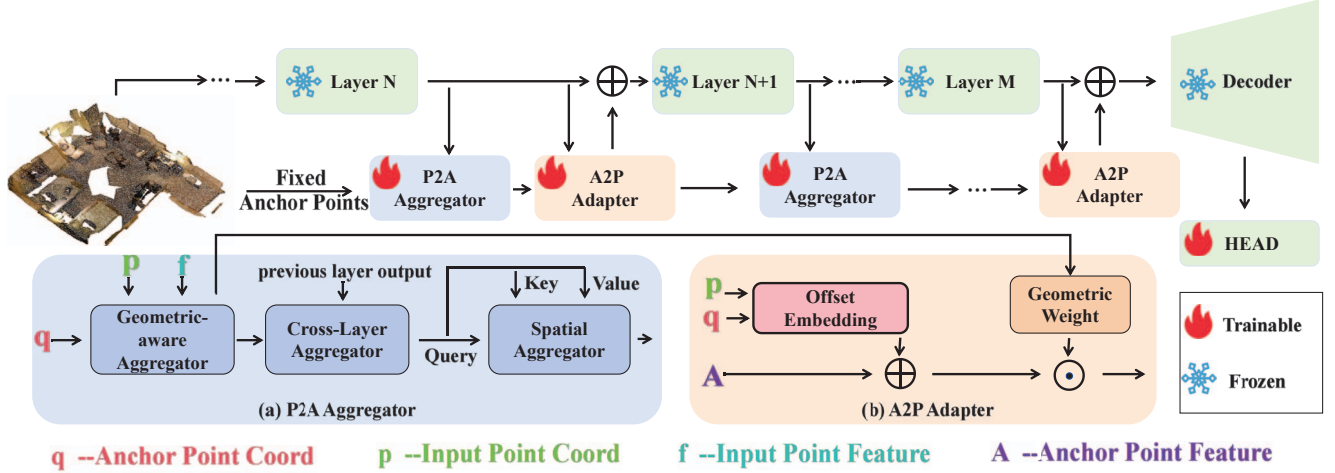


Figure 2. **Illustration of FastAdapter.** FastAdapter can fine-tune any trained point-based model using FPS to adapt it to random sampling. To achieve this, we freeze all parts of the model except for the classification head, and we insert the P2A Aggregator for aggregating local features and the A2P Adapter for supplementing local information.

lost in the random sampling of the input points. Specifically, P2A first introduces a geometry-aware aggregation to extract the local feature. Then, to further refine the local structural features, we designed the Cross-Layer Aggregator and the Spatial Aggregator. The former primarily fuses the features of anchor points across different layers, effectively combining high-level and low-level semantic information, while the latter employs an attention mechanism to merge spatial information between different anchor points, thereby expanding the receptive field.

Geometry-aware Aggregator. We first select L anchor points $q \in R^{L \times 3}$ in the initial scene using FPS, where L is much smaller than the number of input points. Anchor points are fixed throughout the entire forward process of the model. In simple terms, the positions of the anchor points inputted into each layer of FastAdapter remain the same, which facilitates the subsequent cross-layer interaction of information between the anchor points. In our implementation, $L = 100$. At the same time, for the l -th layer of the model M_{FA} , its output (\hat{p}^l, \hat{f}^l) are obtained from Eq. 3. For convenience, we define the output as (p^l, f^l) to serve as the input for FastAdapter.

Next, we inject the point cloud features into the anchor points. To achieve this, we first define the corresponding anchor point index of the i -th point as $j = \arg \min_j \|p_i^l - q_j\|$, which means that we have $p_i^l \in \mathcal{N}_{q_j}$. Then, we calculate the weights for aggregating local features based on the geometric distance from each point to the anchor point.

$$z_i = MLP(p_i - q_j) \quad (4)$$

$$z_j^q = \frac{1}{Len(\mathcal{N}_{q_j})} \sum_{i \in \mathcal{N}_{q_j}} z_i \quad (5)$$

where z_i represents the geometric relationship between i -th point and its corresponding anchor point, while z_j^q represents the local geometric structure information of the j -th anchor point. Then, we calculate the weights for each point.

$$w_i = MLP([z_i, z_j^q]) \quad (6)$$

Finally, we inject the point features into the corresponding anchor points based on the calculated weights.

$$A_{lj} = \frac{1}{Len(\mathcal{N}_{q_j})} \sum_{i \in \mathcal{N}_{q_j}} w_i f_i^l \quad (7)$$

where A_{lj} means the feature of j -th anchor points in l -th layer.

Cross-Layer Aggregator. In this part, we fuse the anchor points' features of the current layer with those of the previous layer and introduce a residual connection. We believe that this approach effectively combines the low-level and high-level semantic features of anchor points, which can enhance the semantic feature representation capability of the anchor points. More importantly, the anchor features from the previous layer can effectively prevent certain local features from being completely discarded during the down-sampling process of the current layer.

$$A_{lj}^{(1)} = MLP([A_{lj}, A_{(l-1)j}]) + A_{lj} \quad (8)$$

It is important to note that in the first layer, the Cross-Layer Aggregator will not be applied since there are no features from a previous layer.

Spatial Aggregator. However, since each point only injects features into the corresponding anchor point, this limits the ability of anchor points to fuse richer features and restricts their receptive field. To address this, we propose the

Table 1. Segmentation results on S3DIS [1] 6-fold, Area-5 and ScanNet [5]. We test our RandPoint on different point-based models and report the mIoU, mAcc, OA, the Throughput (TP) and the GFLOPs (G). Then, **FPS** means Farthest Point Sample, **RS** means Random Sample and **GS** means GridSample.

Methods	Sampling	S3DIS 6-fold		S3DIS Area5		ScanNet	TP(ins./sec.)	GFLOPs
		mIoU	OA	mIoU	OA	mIoU		
KPConv [34]		70.6	-	67.1	-	69.2	-	-
PointTransformer [44]		73.5	90.2	70.4	90.8	70.6	-	5.6
RepSurf [31]		74.3	90.8	68.9	90.2	70.0	-	1.04
ASSANet [27]	FPS	-	-	65.8	88.9	-	-	2.5
PointVector-L [7]		77.4	91.4	71.2	90.8	-	-	10.7
Pix4Point [29]		69.6	89.9	-	-	-	-	-
PointNeXt-L [28]		73.9	89.8	69.5	90.1	69.4	-	15.2
PointNet++ [26]	FPS	54.5	81.0	53.5	83.0	53.5	7.7	30.6
	RS	51.6↓2.9	79.7↓1.3	50.8↓2.7	81.3↓1.7	51.1↓2.4	23	30.6
+FastAdapter	RS	54.0↓0.5	80.6↓0.4	53.1↓0.4	82.7↓0.3	52.9↓0.6	20↑259%	31.8
PointMetaBase-L [18]	FPS	75.6	90.6	69.8	90.6	71.0	7.3	8.48
	RS	73.8↓1.8	89.6↓1.0	68.2↓1.6	89.9↓0.7	67.6↓3.4	27	8.48
+FastAdapter	RS	75.2↓0.4	90.3↓0.3	69.2↓0.6	90.1↓0.5	70.4↓0.6	23↑315%	9.55
PointMetaBase-XL [18]	FPS	76.3	91.0	71.5	91.0	71.8	6.5	39.28
	RS	74.6↓1.7	90.2↓0.8	69.4↓2.1	89.8↓1.2	68.9↓2.9	19	39.28
+FastAdapter	RS	75.8↓0.5	90.7↓0.3	71.2↓0.3	90.9↓0.1	71.3↓0.5	15↑230%	41.80
RandLA [11]	RS	70.2	87.0	68.5	87.4	69.0	23	32.0
+FastAdapter	RS	71.0↑0.8	88.1↑1.1	69.2↑0.7	88.0↑0.6	70.0↑1.0	21	33.1
PTV3 [41]	GS	77.7	-	73.4	-	77.5	25	150
+FastAdapter	GS	78.2↑0.5	-	73.8↑0.4	-	78.0↑0.5	23	155

Spatial Aggregator, which aims to use attention to merge features from different anchor points to expand the receptive field. Additionally, we introduce residual connections to stabilize the training process.

$$A_{lj}^{(2)} = A_{lj}^{(1)} + Attention(A_{lj}^{(1)}, A_{lj}^{(1)}) \quad (9)$$

where *Attention* represents self-attention, which fuses features between different anchor points. Then $A_{lj}^{(2)}$ will be used for the A2P Adapter and as input for the $l + 1$ layer's Cross-Layer Aggregator.

3.3. A2P Adapter

A2P aims to propagate the aggregated local structural features from the anchor points back to the downsampled point cloud in order to supplement the local information lost during random sampling.

Offset Embedding. However, directly backpropagating the same anchor features to different points is suboptimal, as features at different positions should exhibit differences. To address this, we designed Offset Embedding, which predicts the feature offsets based on the positional offsets between the anchors and the points.

$$O_{lj}^i = MLP(p_i - q_j) \quad (10)$$

Then, the features propagated from the superpoint to the i -th point are given by:

$$A_{lj}^i = A_{lj}^{(2)} + O_{lj}^i \quad (11)$$

Geometric Weight. However, the randomness of random sampling may result in varying degrees of local information loss across different regions. We believe that anchor point features should primarily supplement information in regions with significant loss rather than in areas with minimal loss. This approach can help preserve the knowledge initialized by M_{FPS} as much as possible, leading to superior performance. Therefore, we utilize z_i and z_j^q obtained from Eq. 4 and Eq. 5 to calculate the intensity of the superpoint feature propagation to the i -th point. It is important to note that this is different from the meaning of Eq. 6; therefore, we need a new MLP for learning.

$$S_i^l = MLP([z_i, z_j^q]) \quad (12)$$

Finally, we supplement the corresponding points with anchor point features based on the calculated weights.

$$\hat{f}_i^l = f_i^l + S_i^l A_{lj}^i \quad (13)$$

Then \hat{f}_i^l will be used as the input of the $l + 1$ layer.

Table 2. Segmentation results on SemanticKITTI [2]. We report the mIoU and inference speed throughput (TP).

Methods	Sampling	Size	mIoU	TP (ins./sec.)
PointNet [25]	FPS	50K pts	14.6	-
PointNet++ [26]			20.1	-
TangentConv [33]			40.9	-
SqueezeSeg [38]	-	64*2048 pixels	29.5	-
SqueezeSegV2 [39]	-		39.7	-
RangeNet++ [23]	-		52.2	-
RandLA [11]	RS	50K pts	52.9	32
+FastAdapter			53.8	28
PointMetaBase-L	FPS	50K pts	71.8	7.0
	RS		69.2	18
+FastAdapter	RS		71.3	16.7

3.4. Overall Loss

For the input $(X, Y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where x_i and y_i means the i -th point and corresponding label, M_{FA} output the point-wise feature \bar{f}_{FA} and the class predictions P_{FA} .

$$\bar{f}_{FA}, P_{FA} = M_{FA}(X) \quad (14)$$

Then, the segmentation loss is defined as:

$$L_{seg} = CrossEntropy(P_{FA}, Y) \quad (15)$$

Furthermore, to improve the performance of M_{FA} , we additionally introduce feature distillation.

$$\bar{f}_{FPS, -} = M_{FPS}(X) \quad (16)$$

$$L_{distill} = 1 - Cos(\bar{f}_{FA}, \bar{f}_{FPS}) \quad (17)$$

where Cos computes the cosine similarity between the inputs.

Finally, the overall loss can be defined as:

$$L_{overall} = L_{seg} + \lambda L_{distill} \quad (18)$$

It is worth noting that when FastAdapter is embedded into models that already utilize Fast Downsampling methods, we do not use distillation loss; instead, we retain only the segmentation loss.

4. Experiments

4.1. Experimental Setups

We primarily conducted experiments on the indoor datasets S3DIS [1] and ScanNet [5], as well as the large-scale outdoor dataset SemanticKITTI [2], and integrated multiple point-based models to demonstrate the effectiveness of RandPo.

S3DIS. S3DIS is a 3D point cloud dataset for indoor scene understanding. The S3DIS dataset contains large-scale RGB-D data collected from six indoor areas at Stanford

Table 3. Performance Variance Under Different Random Seeds. We used PointMetaBase-XL as the backbone and conducted tests on the S3DIS Area-5 dataset under three different random seeds (1, 42, 1000).

	Backbone _{FPS}	Backbone _{RS}	FastAdapter _{train}	FastAdapter _{infer}
mIoU	±0.4	±1.1	±0.3	±0.1

Table 4. Ablation Study of Fine-tuning Setups. We used PointMetaBase-XL and conducted tests on S3DIS Area-5 dataset.

Setups	mIoU (%)	mACC (%)	OA (%)
Baseline	69.47	75.81	89.87
Head	69.99	75.83	90.43
Decoder+Head	69.53	75.45	90.25
Encoder+Head	70.09	75.88	90.63
Full Fine-tuning	69.42	75.38	90.16
FastAdapter	71.22	77.34	90.86

University, covering a total of 272 rooms with 13 categories of objects.

ScanNet. ScanNet is a large-scale 3D indoor space dataset. It contains 2.5 million views from the indoor environment, and each view has the corresponding camera pose, depth map, and high-quality 3D reconstruction data. In addition, it includes semantic annotations of over 1500 unique scenes covering 20 common indoor categories.

SemanticKITTI. SemanticKITTI is a large-scale outdoor dataset that includes LIDAR scans from 21 sequences, with a total of 43,552 densely annotated scans, each containing 10^5 points. Following common practice, we use sequences 00 to 07 as the training set and sequence 08 as the evaluation set.

4.2. Main Results

Indoor Segmentation. For indoor segmentation tasks, we conducted experiments on S3DIS and ScanNet, integrating FastAdapter into the classic PointNet++ and RandLA as well as state-of-the-art models PointMetaBase-L/XL and PTV3 for testing. In terms of performance, we report mIoU and OA, while for efficiency, we report Throughput (ins./sec) and GFLOPs (G). The efficiency tests were conducted on an NVIDIA GeForce RTX 4090 and two Intel(R) Xeon(R) Gold 6442Y CPUs. The results can be seen in the Table 1,

First, for models PointNet++ and PointMetaBase-L/XL, their original versions use FPS during both training and inference, which significantly hampers inference speed. To address this, we embed FastAdapter into the pre-trained models and replace FPS with random sampling, followed by simple fine-tuning. For each method, we report three types of results: the first is the performance of the pre-trained model using FPS, the second is the result of directly replac-

Table 5. Ablation Study of P2A Aggregator and A2P Adapter. We used PointMetaBase-XL and conducted tests on S3DIS Area-5 dataset.

P2A Aggregator			A2P Adapter		mIoU
Geometric-aware	Cross-Layer	Spatial	Offset	Geometric-weight	
			✓	✓	70.12
✓			✓	✓	70.48
✓	✓		✓	✓	70.85
✓	✓	✓	✓	✓	71.22
✓	✓	✓			69.83
✓	✓	✓	✓		70.68
✓	✓	✓		✓	70.47

ing FPS with random sampling in the pre-trained model, and the third is the outcome of embedding FastAdapter into the pre-trained model and using random sampling to fine-tune. By comparing the first and the second result, we can observe that models using random sampling achieve computation speeds that are several times faster than those using FPS, demonstrating the significant efficiency advantage of random sampling in fast point cloud segmentation. However, due to the limitations of random sampling, its use leads to a substantial decrease in performance. Then, by comparing the second and the third result, we can find that FastAdapter significantly reduces the performance degradation while ensuring superior inference speed, effectively alleviating the issues related to geometric information degradation. Taking the results on ScanNet as an example, when using random sampling, embedding FastAdapter allows PointNet++, PointMetaBase-L, and PointMetaBase-XL to achieve improvements of 1.8%, 2.8%, and 2.4% mIoU, respectively. In comparison to using FPS, this approach only results in a decrease of 0.6%, 0.6% and 0.5% mIoU while significantly enhancing inference speeds by 259%, 315%, and 230%, respectively. More importantly, these models were not specifically designed with architectures for random sampling. By freezing the model parameters and only training FastAdapter, we were able to adapt these models to random sampling. This demonstrates the versatility and effectiveness of FastAdapter. **We have included additional information regarding the training overhead (in terms of epochs and additional parameter) in the supplementary materials.**

Furthermore, for RandLA and PTV3, which are specifically designed with architectures corresponding to their respective fast downSampling methods, embedding FastAdapter during training can further enhance model performance. For example, on ScanNet, FastAdapter improved the mIoU by 1.0% and 0.5% for RandLA and PTV3, respectively. This indicates that FastAdapter is complementary to these specialized architectures and can further address the information loss issues associated with fast downSampling. **Outdoor Segmentation** We also conducted experiments on the large-scale outdoor dataset SemanticKITTI, testing RandLA and PointMetaBase-L to demonstrate the ef-

Table 6. Ablation Study of Training Setups. We used PointMetaBase-XL and conducted tests on S3DIS Area-5 dataset.

Anchor Points Number	50	100	150	200	250	
mIoU	70.87	71.22	71.20	71.13	70.92	
λ	0	1	5	10	100	
mIoU	70.51	71.04	70.92	71.22	70.94	70.51

fectiveness of FastAdapter. The results are shown in Table 2. For RandLA, embedding FastAdapter allows us to achieve an improvement of 0.9% mIoU. In the case of PointMetaBase-L, by replacing FPS with random sampling and integrating FastAdapter for fine-tuning, we can achieve an improvement of 2.1% mIoU compared to directly using random sampling. Furthermore, compared to using FPS, this approach enhances inference speed by 238% while only resulting in a decrease of 0.5% mIoU.

Performance Variance Under Different Random Seeds.

Although Table 1 and Table 2 demonstrate that FastAdapter can effectively assist point-based models in adapting to random sampling, it is important to note that random sampling may yield significantly different results under different random seeds. Here, we aim to explore the stability of FastAdapter across various random seeds. We conducted experiments using PointMetaBase-XL on S3DIS Area 5, with the results shown in Table 3. We selected three different random seeds (1,42,1000) and tested the performance variance of the models under various conditions. First, we reported the variance of the baseline trained from scratch using both FPS (**Backbone_{FPS}**) and RS (**Backbone_{RS}**). The results indicate that, due to the lack of a fixed rule in random sampling, the performance exhibits a considerable variance. Then, we reported **FastAdapter_{train}**, which presents the variance from training FastAdapter under three random seeds. It is evident that, because we fine-tuned a pre-trained model and the FastAdapter effectively compensated for the geometric information lost during the downsampling process, the randomness of random sampling was significantly reduced, resulting in more stable model performance like using FPS. Finally, we reported **FastAdapter_{infer}**, which fixed the random seed to 42 when training model and performed multiple inferences using different random seeds, reporting the performance variance. The results show that the local features captured through anchor point aggregation effectively compensated for the information lost due to random sampling, resulting in stable outcomes.

4.3. Ablation Study

Fine-tuning Setups. Replacing FPS with Random Sample can be considered a distribution shift problem. Therefore, fine-tuning can partially mitigate this issue. Here, we tested different fine-tuning strategies to demonstrate the ef-

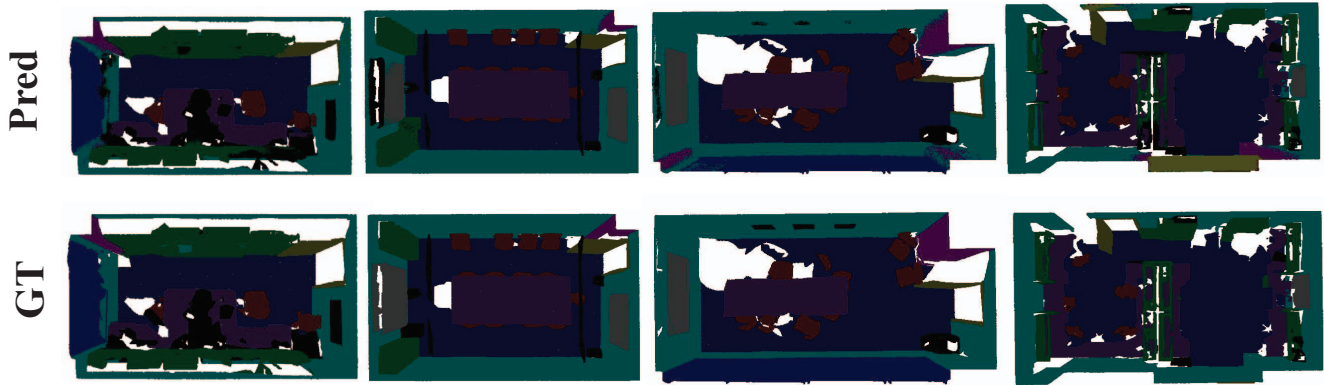


Figure 3. Visualization results on the S3DIS Area 5 dataset. The first row shows the model’s predicted results, while the second row displays the ground truth.

fectiveness of FastAdapter, and the results are shown in Table 4. We conducted experiments using PointMetaBase-XL on S3DIS Area-5 dataset. Firstly, the first row represents the performance of the model during inference when directly replacing FPS with random sampling without any fine-tuning. It is evident that the performance significantly deteriorates due to the geometric information degradation caused by random sampling. Then, we tested the performance of finetuning different parts of the model **without FastAdapter**. The results indicate that over-fine-tuning multiple modules (The rest modules of M_{FPS} except for A2P and P2A) led to suboptimal performance, with full fine-tuning yielding the worst results. In contrast, finetuning the classification head achieved good performance with a smaller computational overhead. We believe this is because freezing most of the parameters helps preserve the knowledge learned during FPS training, thus avoiding the training collapse caused by random sampling. Finally, since FastAdapter continuously aggregates local features through anchor points to supplement information, training only FastAdapter while freezing the weights of the model except for the head can achieve the best results. This indicates that FastAdapter can effectively assist the model in adapting to the distribution resulting from the use of random sampling.

P2A Aggregator and A2P Adapter. Here, we tested the design strategy of the P2A Aggregator and A2P Adapter using PointMetaBase-XL on S3DIS Area-5 dataset, and the results are shown in Table 5. The first four rows are used to test the effect of P2A Aggregator, while the last three rows are used to evaluate the performance of A2P Adapter. First, in the first row, we simply averaged the point features into anchor point without Geometric-aware aggregation. In the second row, we employed a geometry-aware aggregation module that dynamically assigns aggregation weights based on geometric information. As a result, we observed a noticeable performance improvement. Then, by adding the Cross-Layer Aggregator and Spatial Aggregator, the perfor-

mance gradually improved, which demonstrates the effectiveness of the P2A design. Further, In the fifth row, we directly added the anchor point features to the corresponding point cloud without any processing. It can be observed that due to the lack of correction and weighting for the anchor features, the inherent characteristics of the point cloud may be compromised, resulting in a significant decline in performance. In the sixth row, by incorporating Offset Embedding, we can effectively correct the anchor features, preventing biased information from being propagated.

Training Setups. In this section, we primarily tested the impact of the number of superpoints and the distillation loss weight λ on the performance of PointMetaBase-XL on S3DIS Area-5. As shown in Table 6, regarding the number of anchor points, we can see that setting the count to 100 and 150 yields the best performance. Further increasing the number of anchor points does not lead to additional performance improvements; instead, it increases costs. Regarding the weight of the distillation loss, it can be observed that when the weight is set to 0, there is a significant decline in performance.

Visualizations. In Figure 3, we visualized the segmentation results of FastAdapter. It can be observed that FastAdapter can help the model achieve accurate segmentation results under random sampling conditions.

5. Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62376032.

6. Conclusion

In this paper, we introduce FastAdapter, which continuously aggregates local features through anchor points to supplement the information lost during downsampling. This approach effectively alleviates the geometric degradation issues associated with fast downsampling methods.

References

- [1] I Armeni, S Sax, AR Zamir, S Savarese, A Sax, AR Zamir, and S Savarese. Joint 2d-3d-semantic data for indoor scene understanding. arxiv 2017. *arXiv preprint arXiv:1702.01105*. 5, 6
- [2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 6
- [3] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 3
- [4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 3
- [5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 5, 6
- [6] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1201–1209, 2021. 2
- [7] Xin Deng, WenYu Zhang, Qing Ding, and XinMing Zhang. Pointvector: A vector representation in point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9455–9465, 2023. 1, 5
- [8] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017. 1, 2
- [9] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*, 2018. 3
- [10] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 3
- [11] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 2, 3, 5, 6
- [12] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 3
- [13] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1060–1068, 2023. 3
- [14] Shibo Jie, Haoqing Wang, and Zhi-Hong Deng. Revisiting the parameter efficiency of adapters from the perspective of precision redundancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17217–17226, 2023.
- [15] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021. 3
- [16] Zihao Li, Pan Gao, Kang You, Chuan Yan, and Manoranjan Paul. Global attention-guided dual-domain point cloud feature learning for classification and segmentation. *IEEE Transactions on Artificial Intelligence*, 2024. 3
- [17] Dingkan Liang, Tianrui Feng, Xin Zhou, Yumeng Zhang, Zhikang Zou, and Xiang Bai. Parameter-efficient fine-tuning in spectral domain for point cloud learning. *arXiv preprint arXiv:2410.08114*, 2024. 1
- [18] Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17682–17691, 2023. 5
- [19] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 3
- [20] Tao Lu, Chunxu Liu, Youxin Chen, Gangshan Wu, and Limin Wang. App-net: Auxiliary-point-based push and pull operations for efficient point cloud recognition. *IEEE Transactions on Image Processing*, 32:6500–6513, 2023. 2, 3
- [21] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 2
- [22] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8500–8508, 2019. 2
- [23] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 6
- [24] Bohao Peng, Xiaoyang Wu, Li Jiang, Yukang Chen, Hengshuang Zhao, Zhuotao Tian, and Jiaya Jia. Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21305–21315, 2024. 3

- [25] 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, 3, 6
- [26] 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, 3, 5, 6
- [27] 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. 5
- [28] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 5
- [29] Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. 2022. 5
- [30] Haoxi Ran, Wei Zhuo, Jun Liu, and Li Lu. Learning inner-group relations on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15477–15487, 2021. 3
- [31] 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. 1, 3, 5
- [32] Shuofeng Sun, Yongming Rao, Jiwen Lu, and Haibin Yan. X-3d: Explicit 3d structure modeling for point cloud recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2024. 1
- [33] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018. 6
- [34] 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. 3, 5
- [35] Hugues Thomas, Yao-Hung Hubert Tsai, Timothy D Barfoot, and Jian Zhang. Kpconvx: Modernizing kernel point convolution with kernel attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5525–5535, 2024. 3
- [36] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *Advances in neural information processing systems*, 35:14388–14402, 2022. 3
- [37] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022. 3
- [38] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018. 6
- [39] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019. 6
- [40] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 2, 3
- [41] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2, 3, 5
- [42] 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. 2
- [43] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2
- [44] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 3, 5
- [45] Xin Zhou, Dingkan Liang, Wei Xu, Xingkui Zhu, Yihan Xu, Zhikang Zou, and Xiang Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14707–14717, 2024. 3, 1
- [46] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2
- [47] Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*, 2021. 3